

Univerza
v Ljubljani

Fakulteta *za računalništvo
in informatiko*



Dominik Pangeršič

AERODINAMIKA V MODELIH
ZA RAČUNALNIŠKO SIMULACIJO
LETENJA PTIC V JATI

MAGISTRSKO DELO

ŠTUDIJSKI PROGRAM DRUGE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

doc. dr. *Iztok Lebar Bajec*
MENTOR

Ljubljana, 2015

© 2015, Dominik Pangeršič

Rezultati magistrskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov magistrskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

IZJAVA O AVTORSTVU MAGISTRSKEGA DELA

Spodaj podpisani izjavljam, da sem avtor dela, da slednje ne vsebuje materiala, ki bi ga kdorkoli predhodno že objavil ali oddal v obravnavo za pridobitev naziva na univerzi ali drugem visokošolskem zavodu, razen v primerih, kjer so navedeni viri.

S svojim podpisom zagotavljam, da:

- sem delo izdelal samostojno pod mentorstvom doc. dr. Iztoka Lebarja Bajca,
- so elektronska oblika dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko in
- soglašam z javno objavo elektronske oblike dela v zbirki “Dela FRI”.

— Dominik Pangeršič, Ljubljana, oktober 2015.

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Dominik Pangeršič

Aerodinamika v modelih za računalniško simulacijo letenja ptic v jati

POVZETEK

Z nalogo se osredotočamo na preučitev modelov, ki upoštevajo simulacijo aerodinamike letenja ptice z zamahovanjem kril, in analizo možnosti vpeljave teh algoritmov v vedenjske modele letenja ptic v jati, ob izvajanju simulacije v realnem času. S tem namenom smo izdelali podrobnejšo primerjavo med modeli, ki so pri današnji strojni zmogljivosti najbolj primerni za izdelavo celovitega fizikalnega sistema letenja ptic. Na podlagi tega smo predstavili način implementacije, s prednostmi in omejitvami.

Natančneje se z nalogo osredotočimo na predstavitev gibanja ptice, kompleksnosti modela ptice in pristop avtomatske rekonstrukcije raznolikih vrst ptic. Predstavimo model, za katerega z uporabo poenostavljenih enačb računamo aerodinamične sile in predstavimo vpeljavo stabilnih PD krmilnikov. V okviru tega podrobneje opišemo štiri različne, najbolj primerne pristope fizikalno zasnovanih simulacij letenja z zamahovanjem, ki so: a) vizualno prepričljiv model, b) poenostavljen model, c) model z upoštevanjem zračnih tokov in d) podatkovno usmerjeni model. V zaključku predstavimo vpeljavo učnega modela ptice, ki za način krmiljenja v 3D prostoru deluje na podlagi pred-definirane poti in heuristične predstavitve želene orientacije. Z namenom izdelave krmilnika, primerne za vpeljavo v model letenja v jatah, predstavimo metodo združevanja zamahov, brez potrebe hranjenja večje količine podatkov. Zaradi potrebe po izdelavi robustnega krmilnika letenja predstavimo tudi način izdelave podatkovne baze zamahov, namenjene manevriranju pod različnimi vpadnimi koti. Za konec predstavimo možne izboljšave in ideje za nadaljnje delo.

Ključne besede: poenostavljena aerodinamika, PD, SPD, igre, simulacija, skupinsko vedenje, umetno življenje

University of Ljubljana
Faculty of Computer and Information Science

Dominik Pangeršič

Aerodynamics in models for a computer simulation of bird flocking

ABSTRACT

This thesis focuses on examining flapping flight models, and the possibility of their introduction in behavioural algorithms, all while still being able to run in real time. For this purpose, we present a detailed comparison between flapping flight models that are with today's hardware capabilities in mind best suited for the development of a comprehensive physical system of flying birds. We present their implementation specifics as well as the advantages and limitations.

More specifically, we first present an overview of bird flight mechanics, the complexity of 3D bird models suitable for the automatic generation of diverse kinds of birds. We continue by presenting a model for calculating aerodynamic forces with the use of simplified equations and suggest the integration of stable PD controllers. In the context of our problem, we then describe four different, but most appropriate approaches of physically based simulations of flapping flight: a) a visually convincing model, b) a simplified model, c) a model with consideration of airflows and d) a data-driven model. In the discussion part of this thesis, we introduce a type of a learning flapping flight model capable of steering in a 3D space, which works based on a pre-defined path and a heuristic representation of the desired orientation. With the intention of making the controller appropriate for behavioural algorithms, we introduce a procedure for combining strokes, without unnecessarily large data storage. Following the need of a robust flapping controller, we propose a wingbeat database that is able to manoeuvre under different angles of attack. We conclude by presenting possible improvements and ideas for further work.

Key words: simplified aerodynamics, PD, SPD, games, simulations, group behavior, artificial life

ZAHVALA

Zahvaljujem se staršem in tudi vsem ostalim, ki so mi na kakršen koli način pripomogli pri izdelavi pričujočega dela. Še posebej bi se rad zahvalil mentorju Iztoku Lebarju Bajcu za njegovo pomoč in potrpljenje.

Za odzivnost in angažiranost se zahvaljujem tudi Džia-Čiju Vuju in Eundžung Džu, avtorjema vizualno prepričljivega modela in podatkovno usmerjenega modela.

— Dominik Pangeršič, Ljubljana, oktober 2015.

KAZALO

Povzetek	i
Abstract	iii
Zahvala	v
1 Uvod	1
1.1 Skupinsko vedenje	1
1.2 Računalniške igre in filmska industrija	3
1.3 Pregled naloge	4
2 Model ptice	5
2.1 Perje	5
2.2 Skelet in omejitve sklepov	6
2.3 Rep	7
2.4 Gibanje	7
2.5 Predpostavke in poenostavitve modelov	9
3 Aerodinamične sile	13
3.1 Vzgonska in zaviralna sila	14
3.2 Kontrola fizikalnega pogona	16
3.2.1 Proporcionalno-diferencialni krmilnik	18
3.2.2 Stabilen proporcionalno-diferencialni krmilnik	18
4 Pregled obstoječih modelov	21
4.1 Vizualno prepričljiv model	22
4.1.1 Reprezentacija zamaha	22

4.1.2	Aerodinamika	26
4.1.3	Sledenje poti in izračun zelene orientacije	27
4.1.4	Optimizacija parametriziranih zamahov	31
4.2	Poenostavljen model	33
4.2.1	Reprezentacija zamaha	35
4.2.2	Aerodinamika	37
4.2.3	Vpliv tal	39
4.3	Model z upoštevanjem zračnih tokov	39
4.3.1	Reprezentacija zamaha	40
4.3.2	Aerodinamika	42
4.4	Podatkovno usmerjeni model	42
4.4.1	Reprezentacija zamaha	43
4.4.2	Podatkovna baza zamahov	45
5	Razprava	49
5.1	Generiranje in združevanje zamahov	52
5.2	Izdelava podatkovne baze parametrov letenja	55
5.2.1	Polnjenje baze	55
5.2.2	Pridobivanje podatkov iz baze zamahov	56
5.3	Sklep	57

1 Uvod

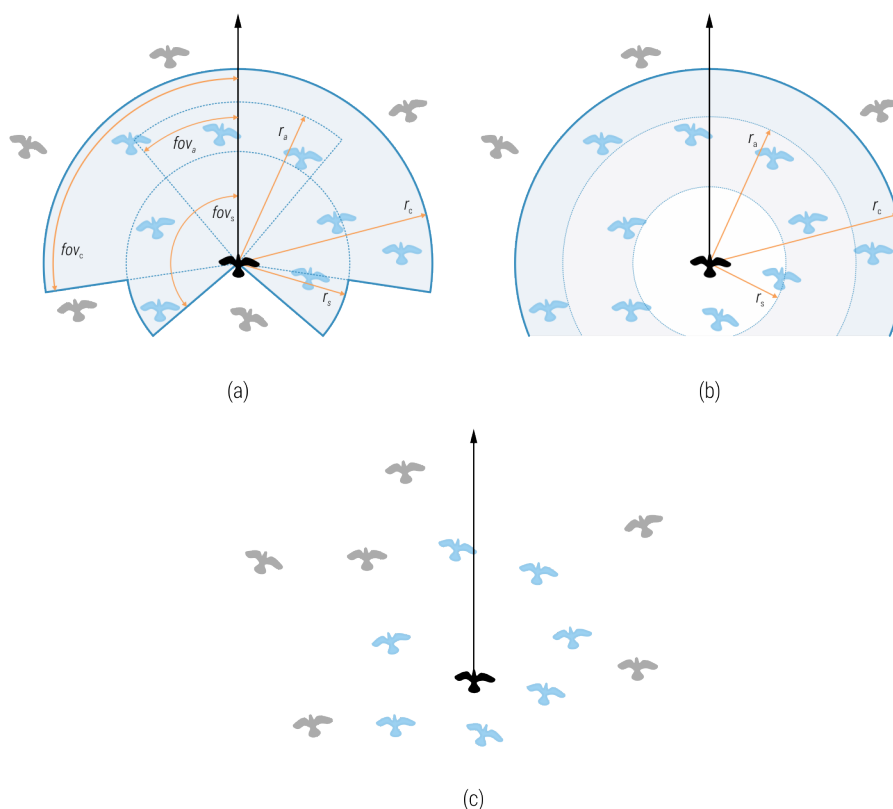
Pri razvoju naprednih tehnologij se razvijalci za navdih vse pogosteje obračajo k naravi [1, 2], saj so strategije, ki jih uporabljajo naravni sistemi [3, 4], plod tisočletij evolucije in prilagajanja delovanja. Primeri takšnih naravnih sistemov so jate ptic in rib, roji čebel, črede, mravlje itd. Ti temeljijo na množici posameznikov, ki opravljajo le preproste operacije, skupen rezultat pa je tako delovanje, ki iz pravil posameznika običajno ni razvidno, npr. usklajeno gibanje.

1.1 Skupinsko vedenje

Kaj je gonilo pri živalih, da se pri vsakdanjih opravilih združujejo v večje skupine in da privede do tako zanimivega vedenja, je eno od glavnih vprašanj raziskovalcev skupinskega vedenja. Velik napredek pri odgovoru na tovrstna vprašanja s pomočjo računalniške tehnologije so v 80-ih letih prejšnjega stoletja opravili Okubo [5], Aoki [6] in Reynolds [7, 8]. Reynolds je prikazal prvi primer vedenjske animacije letenja v jatah (angl. *flocking*), dinamike, ki je v naravi vidna, ko npr. skupina ptic z značilnostmi jate (angl. *flock*) leti, išče hrano in pristaja k počitku. Za doseg takega skupinskega vedenja so potrebne

zgolj tri preproste gonilne sile, ki na osnovi zaznanih sosedov (soseščine) določajo reakcijo obravnavanega bitja (Slika 1.1). Te gonilne sile so:

- razmik (angl. *separation*): preprečevanje prevelikega približevanja drug drugemu,
- poravnava (angl. *alignment*): medsebojno usklajevanje hitrosti in/ali smeri,
- kohezija (angl. *cohesion*): približevanje drug drugemu.



Slika 1.1 (a) Prikazuje Reynoldsov način obravnavanja soseščine, (b) prikazuje conski način obravnavanja soseščine in (c) prikazuje topološko obravnavanje soseščine.

Ko vsaka ptica posebej uravnava med temi tremi gonilnimi silami, v skupini nastane usklajenost pri gibanju, razmikanju, hitrosti in usmerjenosti v času vzleta ali pristanka, kar je biološka definicija jate ptic v letu [9]. Preprostost pristopa je botrovala nastanku različnih modelov, ki obravnavajo skupinsko vedenje, kot so plavanje v jatah (angl. *schooling*) pri ribah [10], zbiranje v rojih (angl. *swarming*) pri žuželkah [11] in zbiranje v čredah (angl. *herding*) pri kopenskih živalih [12–14]. Navkljub številnim modelom je

na področju skupinskega vedenja ostalo še vedno veliko nejasnosti in odprtih vprašanj, kar je tudi vzrok mnogih adaptacij in posodobitev osnovne ideje. Primer posodobitve je način obravnavanja soseščine (Slika 1.1). Reynolds [7] je uporabljal način, kjer je soseščina definirana z vidnim poljem in radijem zaznavanja, a hkrati dopuščal, da ima vsaka gonilna sila svoje parametre. Couzin, et al. [15] so po drugi strani uporabili enotno vidno polje in radij, a gonilne sile obravnavali izključujoče (t. i. conski sistem). S spreminjanjem velikosti cone poravnave so prikazali različna skupinska vedenja, neusklajeno prepletanje (angl. *swarm*), vrtenje okoli praznega središča (angl. *torus*), delno usklajeno gibanje (angl. *dynamic parallel group*) in popolno usklajeno gibanje (angl. *highly parallel group*). Ker sta pristopa omejena z razdaljo, ju s skupnim imenom imenujemo metrična soseščina. Ballerini, et al. [16, 17] so na osnovi visoko ločljivih posnetkov jat predlagali, da skupinsko vedenje ne temelji na metrični soseščini, temveč topološki, kjer posamezna ptica upošteva zgolj sedem osnovnih najbližjih sosedov. Lebar Bajec [3, 18, 19] je po drugi strani z namenom omogočanja lažje razumljivosti obravnavanja gonilnih sil in spodbujanja gradnje modelov s strani etologov vpeljal uporabo mehke logike. V smeri raziskovanja modelov skupinskega vedenja živih organizmov so se tako razvile tri veje raziskovanja. Nekateri zasledujejo minimalistične modele (angl. *self-propelled particles* - *SPP*) predvsem zaradi njihove matematične obvladljivosti [4]. Nekateri želijo modele narediti čimbolj verodostojne, čeprav na račun njihove matematične obvladljivosti [3]. Spet druge pa vodi želja po ohranjanju čim bolj realistične grafike. Kljub vsemu temu mnogi modeli ne upoštevajo zunanjih vplivov in/ali omejitev gibanj. Njihovo pomembnost predstavijo Hildenbrand, et al. [20], ko model letenja v jatah razširijo z uporabo minimalnega upoštevanja aerodinamičnih sil na modelu ptice z nepremičnimi krili in s tem opazno spremembo dinamike gibanja skupine ptic. Na podlagi tovrstnih ugotovitev se tako lahko le sklepa, kakšna bi bila dinamika simulacije ptic v jatah ob upoštevanju podrobnejše upodobitve aerodinamičnih sil, ki delujejo na ptico.

1.2 Računalniške igre in filmska industrija

Prihod računalnikov je na mnogih področjih privedel do velikega preskoka. Reynoldsov model je bil npr. postavljen z namenom avtomatiziranega generiranja animiranih sekvenc rojev netopirjev v filmu *Batman Returns* iz leta 1992. Filmska in hitro rastoča igričarska industrija sta zaradi želje po čim bolj realistični upodobitvi naravnega okolja pripomogli k

razvoju modelov skupinskega vedenja¹. Ob tem pa je zaradi različnih interesov prisoten velik razhod med filmsko in igričarsko industrijo. Filmska industrija še danes strogo stremi k čim bolj realističnim grafičnim učinkom, medtem ko je pri igričarski industriji veliko bolj prisotna želja po izvajanju grafičnih učinkov v realnem času. Sodobne grafične kartice (angl. *GPU*) so odprle nove možnosti, kar je igričarski industriji omogočilo stopiti korak bližje k razvoju simulacij v realnem času, kljub uporabi kar se le da realističnih grafičnih učinkov [21].

1.3 Pregled naloge

Cilj naloge je pregled literature in predstavitev modelov, ki uporabljajo aerodinamične sile za animacijo letenja z zamahovanjem ene same ptice, vse to z namenom vpeljave slednjega v modele skupinskega vedenja. V drugem poglavju tako najprej predstavimo problem, torej letenje ptice in njen računalniški model. V tretjem poglavju predstavimo aerodinamične sile, in kako se te v primeru računalniškega modela izračunavajo. V četrtem poglavju naredimo obširnejši pregled štirih najpomembnejših del; vizualno prepričljivega modela [22], poenostavljenega modela [23], modela z upoštevanjem zračnih tokov [24] in podatkovno usmerjenega modela [25]. V zaključku kritično komentiramo prednosti in slabosti teh modelov ter predstavimo izhodišča za nadaljnje delo.

¹<http://massivesoftware.com/>

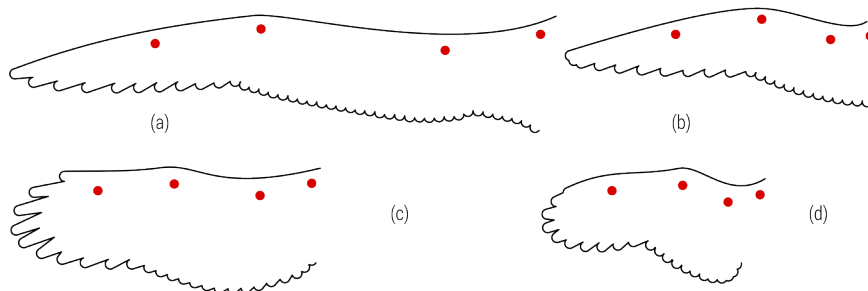
2 Model ptice

Ptiči so (razen netopirjev) edina skupina vretenčarjev, ki lahko v resnici letajo in jadrajo. To jim omogoča edinstvena zgradba njihovega telesa. Čeprav so ptiči največkrat opisani kot vladarji zraka, se veliko časa zadržujejo tudi na kopnem in na vodi. K temu je pri ptici pripomogel razvoj perja. Podobno kot dlaka se je perje razvilo za zaščito pred izgubo toplote in se šele pozneje oblikovalo v trdne letalne ploskve, ki so omogočile letenje. Osnovne telesne in vedenjske poteze ptičev pričajo, da gre za živali, ki so prilagojene za letanje. Mišice so se jim spreminjale tako, da se je krepila telesna moč, predvsem v prsnem delu, telo pa je hkrati postajalo lažje. Kostni so postale votle ali porozne, oblikovane kot tanke ukrivljene plošče.

2.1 Perje

Pero je izrastek kože ptičev, ki ga sestavljajo poroženele celice povrhnjice. Izraščajo se običajno po vsem telesu, razen na kljunu in nogah, in skupaj tvorijo zaščitno plast, ki ji pravimo perje. Ker pokrivajo skoraj celotno telo in so hkrati dobra toplotna izolacija, zmanjšujejo toplotne izgube telesa ter omogočajo vzdrževanje stalno povišane telesne

temperature. Razvoj prednjih okončin v peruti (krila) je ptičem omogočil letenje. Njihova različna oblikovanost (Slika 2.1) in prilagojenost jim omogočata jadranje v velikih višavah, letenje nad oceani, bliskovito lovljenje plena, bežanje pred plenilci in preletavanje velikih razdalj [26, 27].

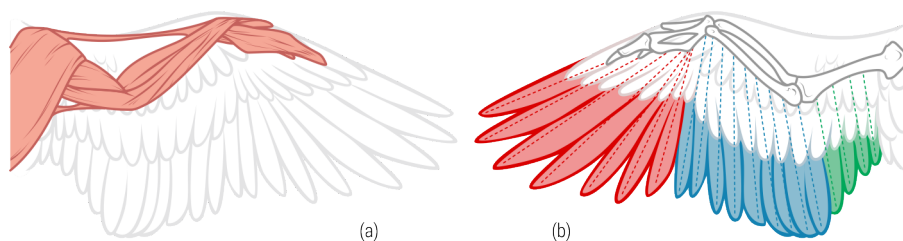


Slika 2.1 Prikaz štirih primarnih tipov kril pri pticah: (a) jadralna, npr. albatros; (b) velike hitrosti, npr. sokol; (c) velik vzgon, npr. orel; (d) hiter vzlet, npr. golob. Z rdečimi pikami so označene lokacije sklepov, ki si od desne proti levi sledijo, ramenski, komolčni, zapestni in dlanski. Povzeto po [26, 27].

Letalna peresa so posebne oblike peres, ki jih najdemo na peruti in repu ptičev in imajo ključno vlogo pri letenju. Letalna peresa na perutih so dolga, toga in nesimetrična. Oblikovana so tako, da med letom omogočajo čim večji vzgon. Glede na njihovo funkcijo jih delimo na primarne, sekundarne in terciarne (Slika 2.2). Repna peresa so pogosto vzdolžno simetrična.

2.2 Skelet in omejitve sklepov

Razvoj letenja s pomočjo mišic je povzročil, da se ptičje okostje bistveno razlikuje od okostij drugih živali. Najbolj opazna posebnost pri ptičih letalcih je močan greben na grodnici, na katerega so pritrjene peruti. Repna peresa rastejo iz trtice. Prednje okončine so povsem prilagojene letenju (Slika 2.2), brez zobe čeljusti pa so se razvile v kljun [28].

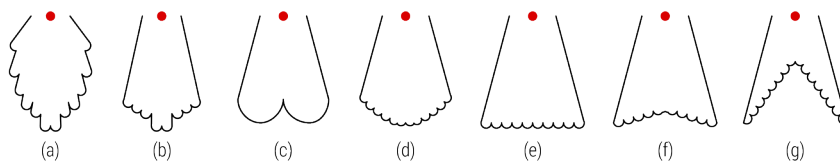


Slika 2.2 (a) Prikaz mišic, napetih na okostje; (b) prikaz delitev in pritrditve peres na skelet; z zeleno so označena terciarna, modro sekundarna in rdečo primarna peresa.

Kosti imajo nizko specifično težo. Dolge kosti ptičev letalcev so votle, toda ojačane z lahкими notranjimi oporami. Golobje okostje tehta na primer samo dvajsetino teže celega telesa [29]. Številni ptiči ne-letalci pa imajo masivne kosti. Edini predel, kjer imajo ptiči več kosti kot drugi vretenčarji, je vrat. Čaplja potrebuje zelo gibljiv vrat, da si lahko ujame hrano in da tudi lahko doseže vsak del telesa, ko si čisti in neguje perje. V čapljinem vratu je 16 do 17 vretenc, v labodjem do 25, sesalci pa imajo samo sedem vretenc.

2.3 Rep

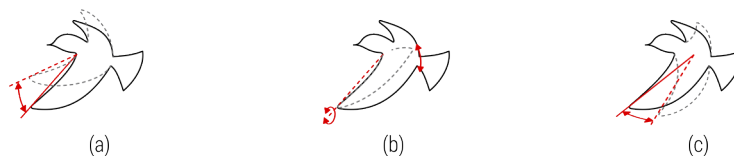
Rep ptice služi predvsem lovljenju ravnotežja in usmerjanju ter v posebnih primerih, kot je počasen let, ustvarjanju dodatnega vzgona. Med ravnim letom razprtje repnih peres poveča vzgon zadnjega dela telesa in rahel spust glave pod ravnino telesa. Zaprtje repnih peres ima nasprotni učinek. Nagib repa povzroči obračanje ptice. Med pristajanjem ptič spusti rep pod telo, kar deluje kot zavora. Podobno kot pri krilih so prilagoditve različnim tipom letenja botrovale nastanku raznolikih oblik repa (Slika 2.3).



Slika 2.3 Prikaz vseh možnih oblik repa: (a) razpršen (b) klinast (c) dvojno-zaobljen (d) zaobljen (e) poravnan (f) zasekan (g) razcepljen. Z rdečo piko je označen repni sklep, oziroma točka pritrdišča repnega perja; povzeto po [26, 27].

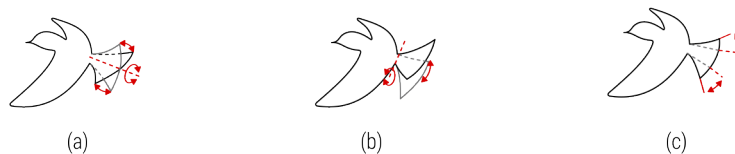
2.4 Gibanje

V primerjavi s fiksnim letalskim krilom se gibanje kril pri ptici razdeli na tri ključne tipe: zamahovanje, zvijanje in uklanjanje naprej-nazaj, kot je prikazano na sliki 2.4. Identično



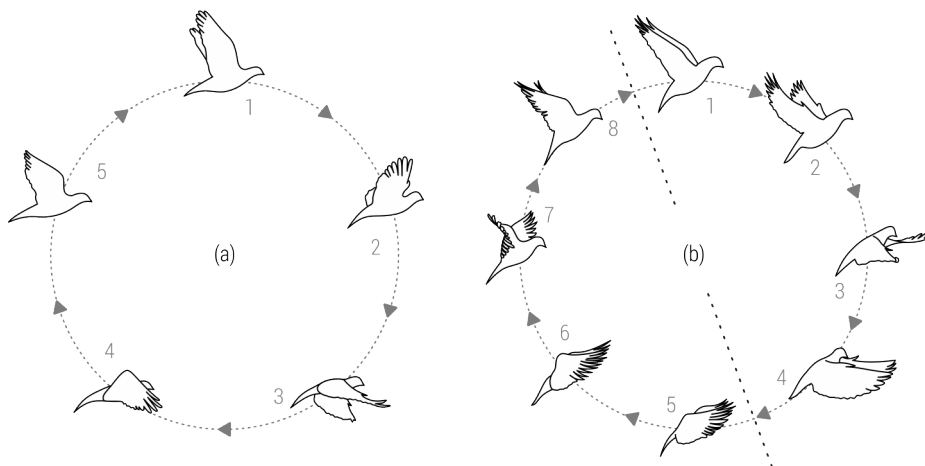
Slika 2.4 Prikaz možnih načinov gibanja peruti ob uporabi zgolj ramenskega, triosnega sklepa: (a) zamahovanje, (b) zvijanje in (c) uklanjanje naprej-nazaj.

kot pri krilih se tudi pri repu tipi gibanja razdelijo na tri podskupine, in sicer: nagibanje, upogibanje in razpiranje, kot prikazuje slika 2.5.



Slika 2.5 Prikaz načinov gibanja repa ob uporabi repnega sklepa in peres: (a) nagibanje, (b) upogibanje in (c) razpiranje.

En cikel zamaha s krili običajno vključuje več sklepov in je sestavljen iz dveh gibov: navzdol, ko se v času letenja naprej krilo giblje dol¹ in navzgor. Slika 2.6a, povzeta iz [30] prikazuje tipičen cikel zamaha goloba pri letu naprej. Skice 1-3 kažejo zamah navzdol, medtem ko 4 in 5 kažeta zamah navzgor. Krilo ustvarja vzgon in potisk pretežno pri zamahovanju naprej-navzdol. Med zamahom navzgor krilo še vedno ustvarja nekaj vzgona ter zelo malo ali nobenega potiska. Upognjenost zunanjega dela krila, kot je vidno na skici 4, je posledica pasivnosti v predelu, kjer je ta povezan na notranji del krila. Tako spodvijanje krila pri zamahu navzgor zmanjšuje upor. Za kratek čas pri zamahu navzgor konice krila zagotavljajo majhno količino pogonske sile, domnevno zaradi zapoznelega gibanja nazaj glede na notranji del krila.



Slika 2.6 (a) Predstavlja en cikel zamaha s krili goloba pri letenju naravnost; (b) predstavlja en cikel zamaha s krili goloba pri počasnem letu; povzeto po [30]; (1b)-(4b) predstavlja zamah navzdol, (5b)-(8b) predstavlja zamah navzgor.

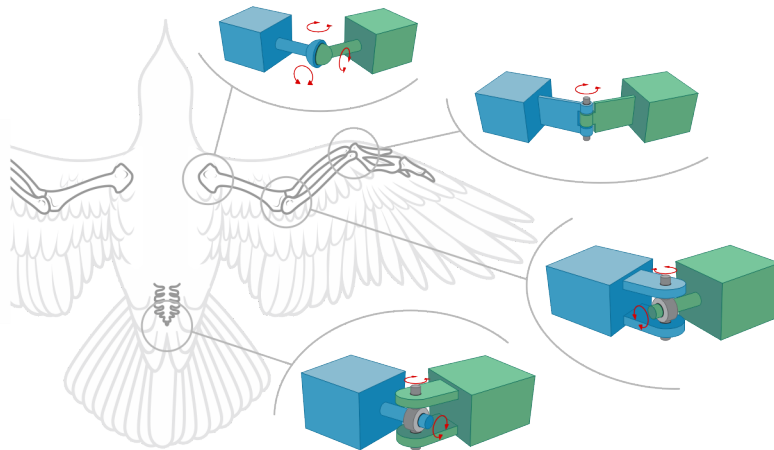
Slika 2.6b prikazuje zamah navzdol in navzgor v počasnem letu. V počasnem letu se vlogi gibanja navzgor in navzdol obrneta. Glavnini vzgonske in potisne sile sta pridobljeni

¹Oziroma naprej pri lebdenju, kar je posebnost kolibrijev.

z zamahom navzgor [30]. Po drugi strani pa zamah navzdol povzroči nekaj vzgona, a brez bistvene pogonske sile. Natančneje pogonska sila med zamahom navzgor nastane iz hitrega, skoraj takojšnjega zasuka in razprtja zunanjega dela krila, kot je tudi vidno na sliki 2.6b, skica 7.

2.5 Predpostavke in poenostavitve modelov

Pravilno delovanje računalniških simulacij, poleg pravilno izgrajenega skeleta in 3D modela, potrebuje tudi vnaprej definirane omejitve vseh sklepov (Slika 2.7). V poročilih so te pogostokrat zamolčane oziroma so nastavljene ad-hoc v dobrobit obravnavanem modelu. Wu & Popović [22] in Parslew [24] npr. pri skeletu ptice upoštevajo tudi omejitve mišic pri razpiranju in zapiranju sklepov, s čimer modelu preprečijo reproduciranje neizvedljivih zamahov v naravi. Ju, et al. [25] se po drugi strani osredotočajo na goloba in ponujajo zadostno število informacij, potrebnih za rekonstrukcijo skeleta (katere so avtorji pridobili z meritvami živega goloba).



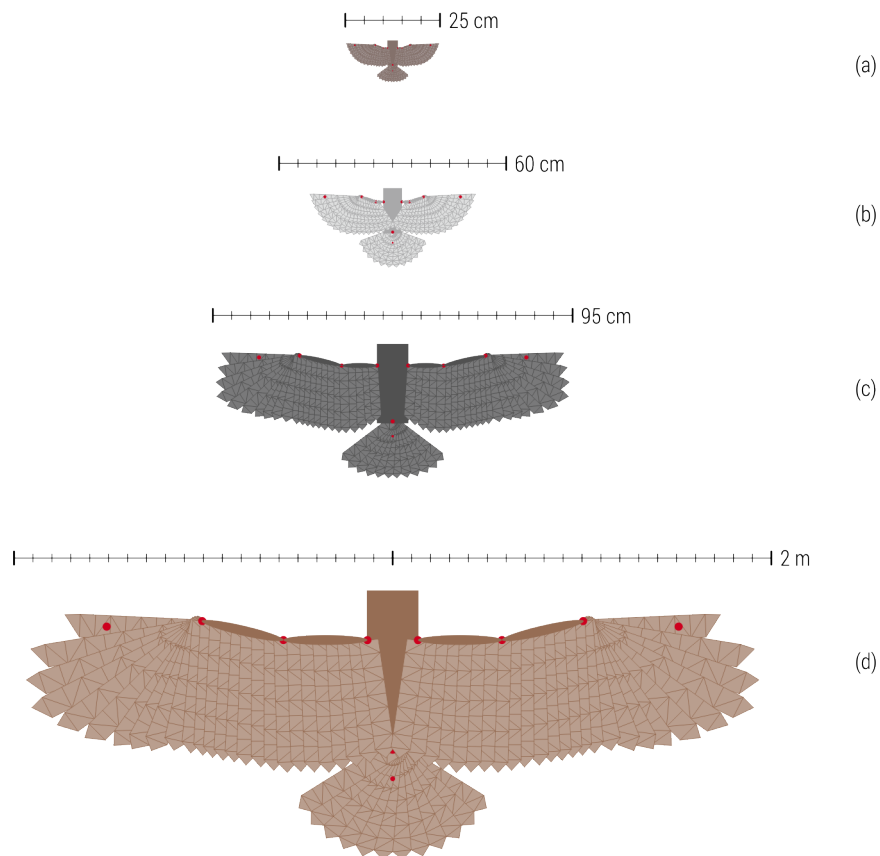
Slika 2.7 Prikaz aproksimacij sklepov in njihove prostorske stopnje (osi vrtenja). Z modrimi kockami so označeni korenski elementi in z zelenimi kockami nasledniki. Ramenski je kroglični, triosni, komolčni in repni dvoosna, zapestni pa enoosni.

Po vzoru Ju, et al. [25] smo zgradili model goloba, kjer je posamezno krilo predstavljeno zgolj z letalnim perjem, in sicer iz dvanaajstih primarnih peres in osmih sekundarnih peres, rep pa predstavlja dvanajest simetričnih peres v dimenzijah, ki so prikazane v tabeli 2.1. Modeliranje vsakega peresa posebej omogoča enostavnejšo programsko rekonstrukcijo raznolikih vrst ptic, le s pravilno podanimi dimenzijami posameznih podskupin peres (Slika 2.8). Poenostavitev modela po vzoru Ju, et al. [25] predpostavlja, da so primarna

Del telesa	Golob				
	w [mm]	h [mm]	d [mm]	m [g]	%
Vrat <i>do</i> Rep	120	50	50	155	60
Zapestje	50	5	100	10	
Komolec <i>do</i> Zapestje	50	5	40	10	15
Rama <i>do</i> Komolec	25	5	20	15	
Rep	30	5	30	25	10
Pero	zunanje [mm]	vmesno [mm]	notranje [mm]	širina [mm]	#
Primarno	160		110	19	10
Sekundarno	110		90	16	8
Terciarno					
Repno	100		100	17	12

Tabela 2.1 Dimenzije členov telesa z masami in intervali dolžin peres goloba. Posamezne osi dimenzij s pogledom na ptico od spredaj so označene kot w širina (angl. *width*), h višina (angl. *height*) in d globina (angl. *depth*). V zgornjem delu tabele % predstavlja delež mase posameznega člena skeleta. V spodnjem delu tabele, so dimenzije peres predstavljene od konice krila do trupa, s # je označeno število peres posamezne podskupine. Povzeto po Ju, et al. [25].

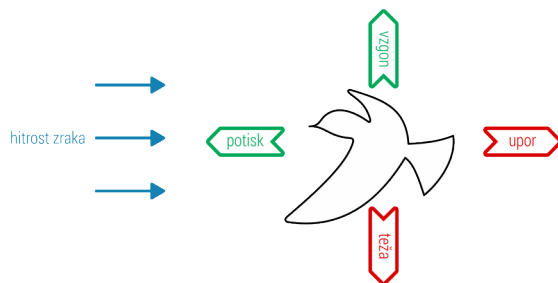
in repna peresa oblikovana v obliki pahljače. Sekundarna (in terciarna) peresa pa so po vzoru Wu & Popović [22] v času odpiranja in zapiranja krila vedno poravnana ob telo ptice. Da bi bili zmožni generiranja bolj verodostojnih modelov, kot sta jih predstavila Wu & Popović [22], smo enostavni linearni porazdelitvi peres Ju, et al. [25] dodali dva korekcijska parametra, dimenzijo vmesnega najdaljšega peresa in indeks njegove pojavitve. S to majhno posodobitvijo je mogoče skonstruirati skoraj vse vrste tipov kril in repov najbolj poznanih ptic, ki so prikazani na slikah 2.1 in 2.3.



Slika 2.8 Prikaz štirih generiranih modelov ptic s prikazanimi maksimalnimi dimenzijami skupnega razpona kril: (a) vrabec, (b) golob, (c) vrana in (d) orel.

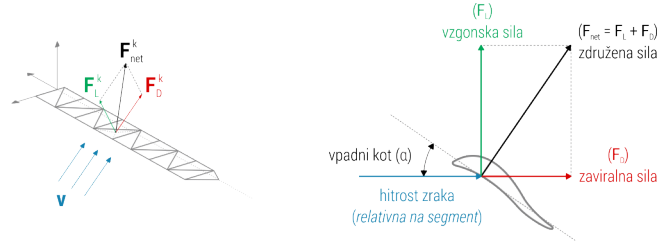
3 Aerodinamične sile

Aerodinamika je pod-področje dinamike tekočin, ki se ukvarja z raziskovanjem gibanja zračnih tokov v interakciji z objekti, kot so letalska krila. Razumevanje gibanja zraka (tokovnega polja) okoli objekta omogoča računanje sil in momentov, ki delujejo na objekt. Sile, s katerimi se srečamo na področju aerodinamike, so: vzgon (angl. *lift*), upor (angl. *drag*), potisk (angl. *thrust*) in teža (angl. *weight*), glej sliko 3.1. Med temi sta vzgon in upor sili, ki nastajata zaradi zračnega toka okoli objekta, npr. ptice oziroma njene peruti. Računanje teh kvantitet je velikokrat zasnovano na podlagi predpostavke,



Slika 3.1 Prikaz ptice in sile, ki se generirajo med letom.

da se tokovno polje vede kot neprekinjeno zvezno gibanje. Neprekinjeno zvezno gibanje tokovnega polja je opisano z lastnostmi, kot so tokovne hitrosti (angl. *flow velocity*), pritisk (angl. *pressure*), gostota (angl. *density*) in temperatura, ki se lahko predstavijo kot funkcije v odvisnosti od pozicije v prostoru in času. Te lastnosti se lahko neposredno ali posredno izmerijo z aerodinamičnimi eksperimenti ali izračunajo z enačbami za pretvorbo mase, momenta in energije v zračnih tokovih. Zaradi računskih omejitev in prevelike kompleksnosti modeliranja celovitega tokovnega polja se v računalniških simulacijah poslužujejo računanja sil večinoma z uporabo izmerjenih lastnosti. Prav tako pogosto predpostavljajo, da se objekt giblje v času mirnega zračnega toka. Da bo telo ostalo v letu, mora sila potiska preglasiti silo upora, sila vzgona pa silo teže. Ptice z zamahovanjem in zvijanjem peruti spreminjajo vpadni kot in s tem proizvajajo tako silo potiska kot silo vzgona.



Slika 3.2 Prikaz računanja vzgonske/zaviralne sile v odvisnosti od vpadnega kota α , relativnega na naravno lego krila.

3.1 Vzgonska in zaviralna sila

Zaradi zagotavljanja skalabilnosti se aerodinamične sile, ki vplivajo na posamezno pero peruti oziroma repa, v našem modelu računa za vsak segment peresa posebej (Slika 3.2). Za izračun sile upora oziroma zaviralne sile in vzgonske sile, ki delujejo na posamezen segment, je treba izračunati vpadni kot segmenta. Ta je pri podani hitrosti zračnega toka relativno na segment, \mathbf{v} , in normali segmenta, \mathbf{n} , izračunan kot

$$\alpha = \arctan\left(\frac{\|\mathbf{v}_n\|}{\|\mathbf{v}_t\|}\right), \quad \mathbf{v}_n = (\mathbf{v} \cdot \mathbf{n})\mathbf{n}, \quad \mathbf{v}_t = \mathbf{n} \times (\mathbf{n} \times \mathbf{v}), \quad (3.1)$$

kjer je \mathbf{v}_n normalna komponenta hitrosti zračnega toka, pravokotna na segment, in \mathbf{v}_t tangentna komponenta hitrosti, vzporedna segmentu. Smer zaviralne sile, \mathbf{d} , in smer vzgonske sile, \mathbf{l} , za segment sta določene z enačbami

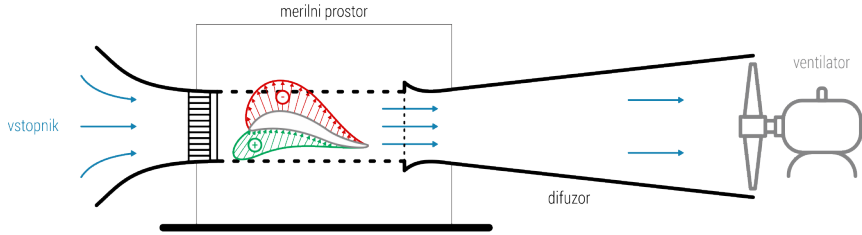
$$\mathbf{d} = \frac{\mathbf{v}}{\|\mathbf{v}\|}, \quad \mathbf{l} = \frac{\mathbf{d} \times \mathbf{n}}{\|\mathbf{d} \times \mathbf{n}\|} \times \mathbf{d}, \quad (3.2)$$

vzgonska in zaviralna sila, ki delujeta na segment se izračunata z enačbama

$$\begin{aligned}\mathbf{F}_L &= \frac{1}{2} C_l(\alpha) \rho S \|\mathbf{v}\|^2 \mathbf{l} \\ \mathbf{F}_D &= \frac{1}{2} C_d(\alpha) \rho S \|\mathbf{v}\|^2 \mathbf{d},\end{aligned}\tag{3.3}$$

kjer je ρ gostota zraka (1.225 kg/m^3), S površina segmenta in $C_l(\alpha)$ ter $C_d(\alpha)$ po vrsti koeficient vzgona in koeficient upora, v odvisnosti od vpadnega kota α .

V aerodinamiki je zelo težko matematično popisati pojave pri gibanju zraka ob letalu. Zato se teoretična obravnavanja dopolnjujejo z izsledki, ki se jih pridobi z meritvami. Navadno se meri sile in momente, ki delujejo na pomanjšan model letala, ki se ga postavi v zračni tok. Najbolj ugodne pogoje za meritve pa je možno doseči le, kadar model letala ali model nekega dela letala (krilo) pritrdimo na letalo naravne velikosti in to merimo kar med letom. Najpogosteje se take meritve opravljajo v vetrnem kanalu. Vetrni kanal je naprava, v kateri se s pomočjo ventilatorja ustvari zračni tok. V merilni prostor vetrnega kanala (Slika 3.3) se na posebno tehtnico pritrdi model letala ter z njo meri sile (vzgon, upor) in momente, ki delujejo nanj.

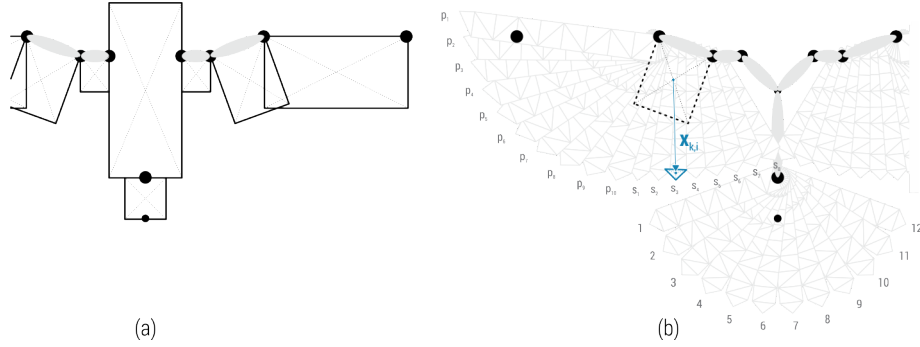


Slika 3.3 Prikaz vetrnega kanala s krilnico. Na krilnici je s poljem zelenih puščic prikazan nadtlač in s poljem rdečih puščic podtlač.

Gibanje krila ptice oziroma spreminjanje vpadnega kota posledično privede do zveznega spreminja vzgonske in zaviralne sile. Modeli, kot je [23], spreminjanje koeficientov upora oziroma vzgona v odvisnosti od vpadnega kota poenostavijo z uporabo linearnih zlepkov in s tem rušijo koncept zveznega spreminjanja sil, ki delujejo na perut. Wu & Popović [22] zato uporabljata kosinusne zlepke, Parslew [24] kar preproste sinusne funkcije, Ju, et al. [25] pa kubične interpolacijske zlepke (angl. *Catmull-Rom spline*).

Ker se v modelu aerodinamične sile računajo za vsak segment posebej, je treba izračunati skupno silo in navor, ki delujeta na sklep. To se izračuna z enačbami:

$$\mathbf{F}^j = \sum_k \mathbf{F}_{net}^k, \quad \boldsymbol{\tau}^j = \sum_k \mathbf{x}_{k,i} \times \mathbf{F}_{net}^k, \quad \mathbf{F}_{net}^k = \mathbf{F}_L^k + \mathbf{F}_D^k, \tag{3.4}$$



Slika 3.4 (a) Prikaz fizikalnega objekta ptice s kvadri kot center mase; (b) prikaz izračuna momentnega vektorja od centra mase podskupine peres i do centra segmenta na peresu k .

kjer je \mathbf{F}_{net}^k skupna sila segmenta, \mathbf{F}^j skupna sila, ki deluje na pero j , τ^j skupni navor, ki ga ustvarja pero j , in $\mathbf{x}_{k,i}$ momentni vektor oziroma vektor od centra mase podskupine peres i (oziroma center rotacije) do centra segmenta k na peresu j (Slika 3.4b). Za izračun aerodinamične sile, ki deluje na posamezno podskupino peres oziroma sklep, je treba izračunati:

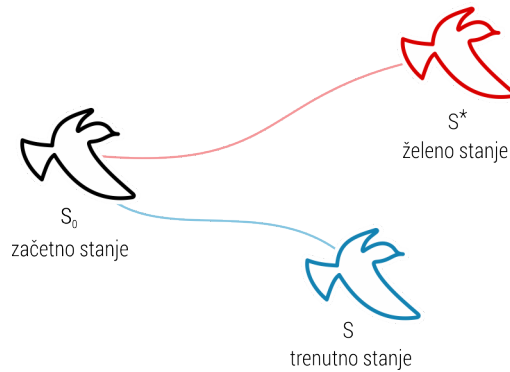
$$\mathbf{F}_{aero}^i = \sum_j \mathbf{F}^j, \quad \tau_{aero}^i = \sum_j \tau^j, \quad (3.5)$$

kjer sta \mathbf{F}_{aero}^i in τ_{aero}^i v tem vrstnem redu aerodinamična sila in navor podskupine peres oziroma sklepa. Poleg teh dveh je s \mathbf{F}_g gravitacijsko silo zagotovljeno apliciranje pojavov iz okolja na fizikalni skelet. Za omogočanje krmiljenja slednjega je tako treba le še pravilno izračunati potrebne težnje (sile/navore), s katerimi se model lahko krmili.

3.2 Kontrola fizikalnega pogona

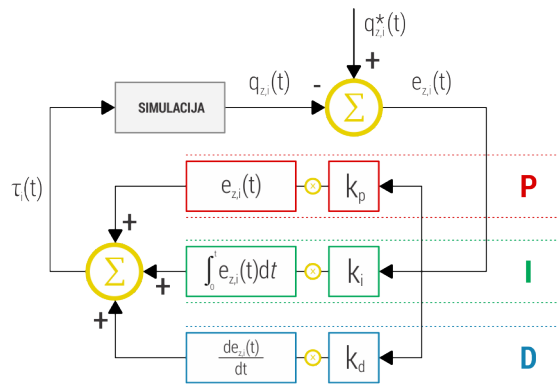
V zadnjem času je zaradi pohitritev osebnih računalnikov v porastu veliko fizikalnih simulacij in animacij [31], katerih največji izziv je zagotavljanje numerične stabilnosti (Slika 3.5). Veliko modelov, kot sta Wu & Popović [22] in Ju, et al. [25] zaradi zelo enostavne implementacije v ta namen uporablja t. i. proporcionalno-diferencialne krmilnike (angl. *Proportional-Derivative Controllers*) oziroma krajše PD krmilnike.

Osnovni koncept PD krmilnikov v fizikalnih animacijah je izračun navora sklepa, $\tau_i(t)$, ki je linearno proporcionalen razliki trenutnega stanja sklepa, $q_{z,i}(t)$, in želenega stanja sklepa, $q_{z,i}^*(t)$. Stabilnost že tako nestabilnega krmilnika pa se lahko poruši že samo z npr. apliciranjem gravitacije, kar privede do tega, da se v naslednji iteraciji stanje simulacije



Slika 3.5 Prikaz delovanja PD krmilnika iz izračunanega želenega stanja S^* do končnega trenutnega stanja S . Za zagotavljanje stabilne simulacije, mora biti razhod med tema stanjema čim manjši.

razlikuje od želenega. Za preprečevanje tega se v industriji velikokrat uporabljajo PID krmilniki (angl. *Proportional-Integral-Derivative Controllers*) (Slika 3.6), pri katerih se upošteva tudi integracijsko napako. Z integralnim delom se kompenzirajo vse predhodne napake, ki bi morale biti tekom simulacije odpravljene. Navkljub zelo enostavni rešitvi, v fizikalnih animacijah taka posodobitev ni primerna, ker se pri vsaki spremembi želenega stanja izniči integracijski del krmilnika, kar privede do še veliko bolj nestabilne animacije kot le z uporabo PD krmilnika.



Slika 3.6 Prikaz zanke PID krmiljenja. $q_{z,i}(t)$ predstavlja trenutno stanje simulacije in $q_{z,i}^*(t)$ želeno trenutno stanje. Iz obeh skupaj se dobi $e_{z,i}(t)$ napako med simulacijo in želenim stanjem. Po krmiljenju s kontrolnimi parametri k_p , k_i in k_d se pridobi korekcijsko vrednost $\tau_i(t)$, s katero se privede simulacijo do želenega stanja.

3.2.1 Proporcionalno-diferencialni krmilnik

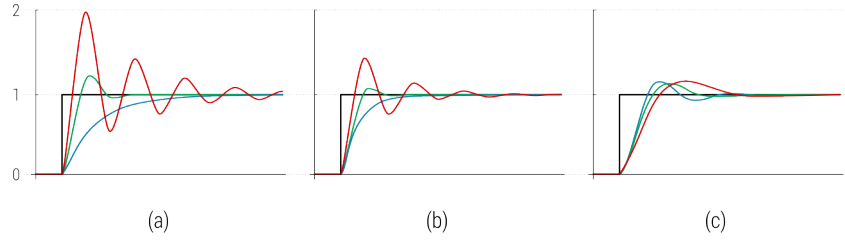
Osnovna enačba za PD krmilnik je

$$\tau_i(t) = k_p(q_{z,i}^*(t) - q_{z,i}(t)) - k_d\dot{q}_{z,i}(t), \quad (3.6)$$

kjer sta k_p in k_d v tem vrstnem redu proporcionalni in diferencialni regulacijski koeficient. S to enačbo se računa navor na podlagi želenega stanja $q_{z,i}^*(t)$ in trenutnega stanja $q_{z,i}(t)$ z upoštevanjem le trenutne hitrosti gibanja $\dot{q}_{z,i}(t)$. Za zmanjšanje oscilacije sledenja se v enačbo vključi tudi želeno hitrosti $\dot{q}_{z,i}^*(t)$ kot neničelno referenčno vrednost, s čemur se dobi enačbo

$$\tau_i(t) = k_p(q_{z,i}^*(t) - q_{z,i}(t)) + k_d(\dot{q}_{z,i}^*(t) - \dot{q}_{z,i}(t)), \quad (3.7)$$

najbolj znano na področju na fiziki temelječih animacij. Žal je PD krmilnik težko obvladljiv (Slika 3.7) in močno odvisen od koeficientov k_p in k_d .



Slika 3.7 Prikaz odziva P, I, D glede korekcijskih parametrov k_p , k_i in k_d . Na grafih so s črno obarvanimi krivuljami prikazane hitre spremembe stanja sistema, rdeče obarvane krivulje prikazujejo odziv pri vrednosti korekcijskega parametra 2, zelena krivulja 1.1 in modra krivulja 0.5; (a) prikaz proporcionalnega odziva sistema; (b) prikaz integralnega odziva sistema; (c) prikaz diferencialnega odziva sistema.

3.2.2 Stabilen proporcionalno-diferencialni krmilnik

Namesto uporabe trenutnega stanja $q_{z,i}(t)$ se pri stabilnih proporcionalno-diferencialnih krmilnikih (angl. *Stable Proportional-Derivative Controllers*) oziroma krajše SPD krmilnikih [32] za izračun navora uporabi naslednje stanje $q_{z,i}(t+1)$:

$$\tau_i(t) = k_p(q_{z,i}^*(t+1) - q_{z,i}(t+1)) - k_d\dot{q}_{z,i}(t+1). \quad (3.8)$$

Ker se običajno ne ve, kakšno je naslednje stanje, se z uporabo Taylorjeve vrste razširi $q_{z,i}(t+1)$ in $\dot{q}_{z,i}(t+1)$. Za razširitev se uporabi zgolj prva dva člena vrste, kar da:

$$\begin{aligned} q_{z,i}(t+1) &= q_{z,i}(t) + \dot{q}_{z,i}(t)\Delta t \\ \dot{q}_{z,i}(t+1) &= \dot{q}_{z,i}(t) + \ddot{q}_{z,i}(t)\Delta t. \end{aligned} \quad (3.9)$$

Podobno kot v primeru PD krmilnikov se lahko tudi v primeru SPD krmilnikov želena hitrost uporabi za zmanjšanje oscilacije sledenja, kot kaže enačba (3.10).

$$\tau_i(t) = k_p(q_{z,i}^*(t+1) - q_{z,i}(t) - \dot{q}_{z,i}(t)\Delta t) + k_d(\dot{q}_{z,i}^*(t+1) - \dot{q}_{z,i}(t) - \ddot{q}_{z,i}(t)\Delta t) \quad (3.10)$$

SPD krmilnik je podobno kot PD krmilnik še vedno težko nastavljen, a vendarle lažje. Slednje predvsem zato, ker lahko z optimizacijo enega od koeficientov in ohranjanjem odvisnosti $k_d \geq k_p\Delta t$, zagotovimo stabilnost krmilnika. Ena od slabosti posodabljanja iz PD v SPD krmilnik je le ta, da vsi koeficienti, ki so bili prirejeni za PD krmilnik, niso uporabni v SPD krmilniku in obratno. Uporaba SPD krmilnikov pa je s tem še vedno neskalabilna, a vendarle veliko bolj stabilna kot pri konvencionalnih PD krmilnikih.

4 Pregled obstoječih modelov

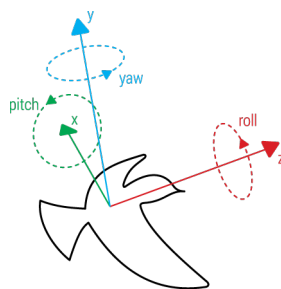
Zaradi lažjega razumevanja in boljšega pregleda nad primerljivimi modeli bomo definirali standardno stanje simulirane ptice. Da bi bilo stanje čim bolj poenostavljeno, so vse prostostne stopnje sistema združene v en sam vektor, \mathbf{q} . Ta vektor tako \mathbf{q}_p opredeljuje pozicijo, \mathbf{q}_r orientacijo ptice ter \mathbf{q}_z stanja vseh sklepov skeleta. Simulacijsko stanje \mathbf{S} je tako podano s stanjem ptice v danem trenutku, \mathbf{q} , in hitrostjo spreminjanja stanja - prvim odvodom, $\dot{\mathbf{q}}$:

$$\mathbf{S} = \begin{bmatrix} \mathbf{q} & \dot{\mathbf{q}} \end{bmatrix}. \quad (4.1)$$

Izračun naslednjega stanja sistema, izhajajoč iz katerega koli želenega stanja v času t , potrebuje le \mathbf{S}_0 začetno stanje simulacije, \mathbf{F} zunanje sile (gravitacija in aerodinamične sile) in navore sklepov skeleta $\boldsymbol{\tau}(t)$:

$$\mathbf{S}(\mathbf{S}_0, \mathbf{F}, \boldsymbol{\tau}(t), t) = \begin{bmatrix} \mathbf{q}(t) & \dot{\mathbf{q}}(t) \end{bmatrix}. \quad (4.2)$$

Orientacijo simulirane ptice v prostoru podajamo prek lokalnega koordinatnega sistema, kjer so osi razporejene po vzoru Reynoldsa [7], kot prikazuje slika 4.1.



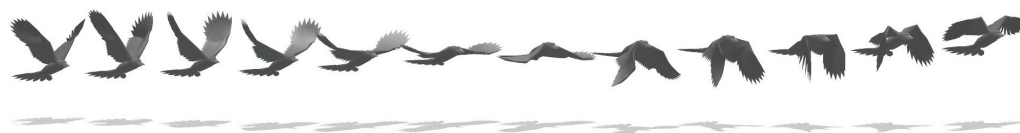
Slika 4.1 Prikaz orientacije ptice q_r , z uporabo desno-sučnega koordinatnega sistema, pri katerem je smer letenja naprej z-os, smer gor y-os in smer levo x-os.

4.1 Vizualno prepričljiv model

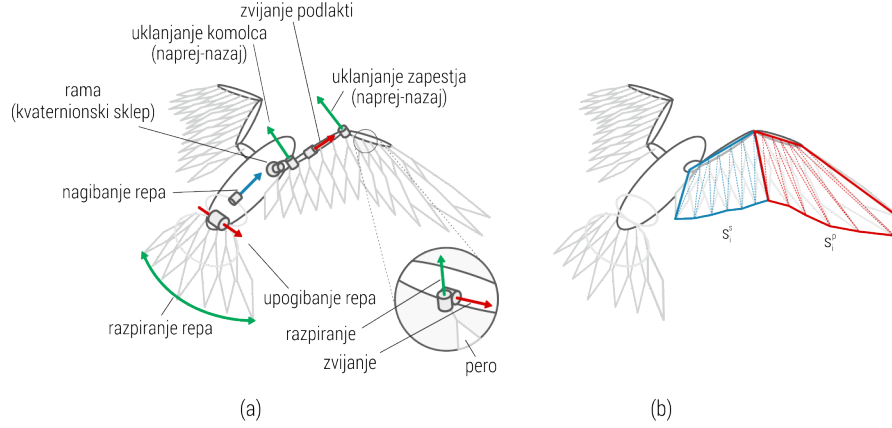
Wu & Popović [22] sta predstavila model, ki je hkrati eden od prvih modelov animacije letenja z zamahovanjem (Slika 4.2) in hkrati prvi, ki ob uporabi fizikalnega pogona modelira polno funkcionalen skelet ptice (Slika 4.3). Ramakrishnananda & Wong [33], ki sta že leta 1999 predstavila model animacije letenja z zamahovanjem, sta uporabila močno poenostavljen model in parametre modela nastavljala ročno. Wu & Popović [22] sta z željo po avtomatskem generiranju animacije letenja (po predhodno definirani poti), v ta namen predstavila način optimizacije parametrov posameznega zamaha z uporabo algoritma simuliranega ohlajanja.

4.1.1 Reprezentacija zamaha

Velikost seta parametrov v učnem modelu predstavlja stopnjo preiskovalnega prostora optimizacijskega algoritma. Zato je pomembno, da se nastavi čim manjše število parametrov (Tabela 4.1), s katerimi se lahko zagotavlja zadostne manevrske sposobnosti modela. Z željo po čim boljšem in hitrejšem optimiziranju zamahov Wu & Popović [22], predstavit en zamah ptice kot vektor parametrov, ki ga bomo označevali z \mathbf{u} . Vodenje večjega dela parametrov, gledano po prostostnih stopnjah sklepov, je razdeljeno na dva dela: zamah navzdol in zamah navzgor, ki sta v tem vrstnem redu označena z nadpi-



Slika 4.2 Prikaz enega cikla zamaha orla med vzletanjem. Povzeto po Wu & Popović [22].



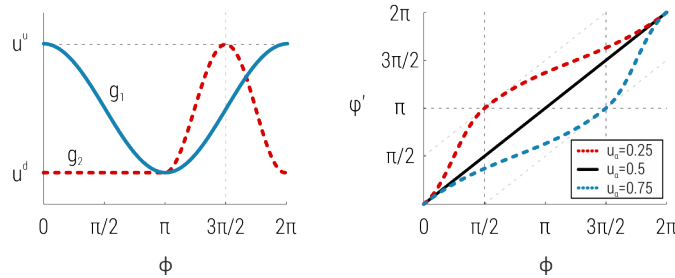
Slika 4.3 (a) Prikaz konstrukcije skeleta ptice. Prikazan model ima 3 prostostne stopnje za rama, 2 prostostni stopnji za komolec, 1 prostostno stopnjo za zapestje in 2 prostostni stopnji za rep. (b) Prikazan model z razdelitvijo krila na 2 površini peres; s_i^p ki predstavlja površino primarnih peres, zunanji del krila, in s_i^s ki predstavlja površino sekundarnih peres, notranji del krila.

som d in u . Posamezen parameter je v intervalu $[0, 1]$, njegovo časovno spreminjanje pa določata funkciji

$$g_1(u_i^d, u_i^u, \phi) = (u_i^u - u_i^d) \frac{1 + \cos(\phi)}{2} + u_i^d, \quad (4.3)$$

$$g_2(u_i^d, u_i^u, \phi) = \begin{cases} (u_i^u - u_i^d) \frac{1 - \cos(2\phi)}{2} + u_i^d & \pi \leq \phi < 2\pi \\ u_i^d & 0 \leq \phi < \pi \end{cases}, \quad (4.4)$$

ki predstavljata ciklično zvezni funkciji, prikazani na sliki 4.4. Na podlagi opazovanj dinamike letenja se funkcija g_1 uporablja za spreminjanje kotov $q_{z,1}^*$ zamahovanja v rami in $q_{z,7}^*$ upogibanje repa, za spreminjanje ostalih pa se uporablja g_2 . Funkciji sta definirani tako, da je ob $\phi = 0$ sklep v začetni fazi zamaha dol, ob $\phi = \pi$ sklep v začetni fazi zamaha gor in ob $\phi = 2\pi$ sklep v končni fazi zamaha gor.



Slika 4.4 LEVO Prikaz funkcij računanja kotov g_1 in g_2 (x-os radiani, y-os faktor), DESNO fazna transformacije za omogočanje spremembe razmerja časa gibanja krila gor in dol (x-os radiani, y-os radiani).

Parameter	k	Stran	Opis
T			Trajanje zamaha
u_α, u_β			Regulacijske spremenljivke zamaha
u_1^d, u_1^u	1	L/D	Rama kot zamahovanja
u_2^d, u_2^u	2	L/D	Rama kot zvijanja
u_3^d, u_3^u	2	L/D	Rama kot uklanjanja naprej-nazaj
u_4^d, u_4^u	2	L/D	Komolec kot zvijanja
u_5^d, u_5^u	2	L/D	Komolec kot uklanjanja naprej-nazaj
u_6^d, u_6^u	2	L/D	Zapestje kot uklanjanja naprej-nazaj
u_7^d, u_7^u	1		Rep kot upogibanja
u_8			Rep kot razpiranja
u_9			Rep kot nagibanja

Tabela 4.1 Prikaz vektorja parametrov zamaha \mathbf{u} . Število vseh parametrov je 32, od tega dva korekcijska in en čas.

Računanje sprememb kotov sklepov

Za izračun navorov sklepov $\tau_i(t)$, ki privedejo do spreminjanja sklepov model uporablja dinamiko naprej (angl. *forward dynamics*) oziroma PD krmilnik, predstavljen v 3. poglavju. Za izračun novega želenega stanja, \mathbf{q}_z^* , se izvede linearna interpolacija parametrov zamaha, \mathbf{u} , iz intervala $[0, 1]$ v interval razpona posameznega sklepa (Tabela 4.2), katerih Wu & Popović [22] ne navedeta.

	$q_{z,1} [^\circ]$	$q_{z,2} [^\circ]$	$q_{z,3} [^\circ]$	$q_{z,4} [^\circ]$	$q_{z,5} [^\circ]$	$q_{z,6} [^\circ]$	$q_{z,7} [^\circ]$	$q_{z,8} [^\circ]$	$q_{z,9} [^\circ]$
max	40	95	20	30	170	-20	45	70	25
min	-85	-25	-65	-120	20	-170	-45	10	-25

Tabela 4.2 Prikaz omejitev sklepov goloba, pridobljene s pomočjo modela Ju, et al. [25]. Koti se obravnavajo relativno na korenski sklep, pri čemer se upošteva, da je ramenski sklep pravokotno poravnan s telesom.

Za parametre, ki se delijo na zamah navzdol in zamah navzgor, se uporabi

$$q_{z,i}^*(t) = \underline{q}_{z,i} + (\bar{q}_{z,i} - \underline{q}_{z,i}) g_k(u_i^d, u_i^u, \phi(t)) \quad (4.5)$$

za ostale konstantne parametre pa

$$q_{z,i}^*(t) = \underline{q}_{z,i} + (\bar{q}_{z,i} - \underline{q}_{z,i}) u_i, \quad (4.6)$$

kjer je k podan v tabeli 4.1, $\underline{q}_{z,i}$ in $\bar{q}_{z,i}$ v tem vrstnem redu pa predstavljata minimalno in maksimalno dovoljeno mejo gibanja sklepa. Za generiranje bolj raznolikih zamahov

se namesto $\phi(t) = \frac{2\pi t}{T}$, kjer je T dolžina zamaha v časovnih korakih in t trenutno trajanje enega cikla zamaha, uvede funkcija $h(\phi(t), u_{\alpha|\beta})$. Funkcija je fazna transformacija, namenjena spreminjanju intenzitete zamaha gor/dol z uporabo korekcijskih parametrov u_α in u_β , ki se v tem vrstnem redu uporabljata s funkcijo g_1 in funkcijo g_2 . Za izračun slednje se uporabi enačbo

$$h(\phi, u_{\alpha|\beta}) = \begin{cases} \phi - \pi u_{\alpha|\beta} + \frac{\pi}{2} + \frac{\pi}{2}(2u_{\alpha|\beta} - 1) \cos \frac{\phi}{2u_{\alpha|\beta}} & \phi < 2\pi u_{\alpha|\beta} \\ \phi - \pi u_{\alpha|\beta} + \frac{\pi}{2} + \frac{\pi}{2}(1 - 2u_{\alpha|\beta}) \cos \frac{\phi - 2\pi u_{\alpha|\beta}}{2 - 2u_{\alpha|\beta}} & \text{ostalo} \end{cases}, \quad (4.7)$$

ki je definirana na intervalu $[0, 2\pi]$. Za zagotavljanje pravilne transformacije je treba upoštevati, da je funkcija stabilna le z uporabo $u_{\alpha|\beta}$ na intervalu $[0.25, 0.75]$. Za zagotovitev stabilnejše fizikalne simulacije ter ohranitev sinhronizacije zamaha tudi po fazni transformaciji je treba poskrbeti, da se funkcija g_2 začne z zamahom navzdol. To pomeni da se ϕ , ki nastopa v enačbah g_1 in g_2 , zamakne za $\pi(1 + u_{\alpha|\beta})$, tako da se konec zamaha gor/začetek zamaha dol funkcije g_2 ujema s tistim iz funkcije g_1 . To se najlažje doseže kar z razširitvijo fazne transformacije

$$h^t(\phi, u_{\alpha|\beta}) = h((\phi + \pi(1 + u_{\alpha|\beta})) \bmod 2\pi, u_{\alpha|\beta}) \quad (4.8)$$

Glede na fazno transformacijo zamaha in njeno odvisnost od parametrov $u_{\alpha|\beta}$ za gibanje gor velja $\nu(\phi, u_{\alpha|\beta}) = \pi(1 + u_{\alpha|\beta}) < \phi \leq \pi(1 - u_{\alpha|\beta})$. Pravilno delovanje enačbe pa je zagotovljeno le pri pogoju, da je $u_i^d < u_i^u$. V nasprotnem primeru je treba rezultat $\nu(\phi, u_{\alpha|\beta})$ negirati.

Z namenom poenostavitve učnega modela se para parametrov u_5 in u_6 obravnavata kot skupen parameter $u_{5,6}$ kar pomeni, da se npr. z izračunom u_6 posledično nastavi tudi u_5 s predpostavko, da sta meji sklepov obratno sorazmerni. Avtomatsko odpiranje in zapiranje krila se izračuna

$$q_{z,5}^*(t) = (q_{z,6}^*(t) - \underline{q}_{z,6}) \frac{\underline{q}_{z,5} - \bar{q}_{z,5}}{\bar{q}_{z,6} - \underline{q}_{z,6}} + \bar{q}_{z,5}. \quad (4.9)$$

Iz te poenostavitve sledi tudi linearno spreminjanje meje stanja nadlakti, u_4 , v odvisnosti para parametrov $u_{5,6}$, ki se izračuna

$$\bar{q}'_{z,4} = \bar{q}_{z,4} + \frac{(q_{z,6}^*(t) - \underline{q}_{z,6})(\underline{q}_{z,4}\chi - \bar{q}_{z,4})}{\bar{q}_{z,6} - \underline{q}_{z,6}}, \quad (4.10)$$

kjer je $\bar{q}'_{z,4}$ korigirana maksimalna meja zvijanja komolca in $\chi \in [0, 1]$ faktor, s katerim se zagotovi, da ob zaprtju krila podlaket privede do rotacije zapestja, poravnano proti

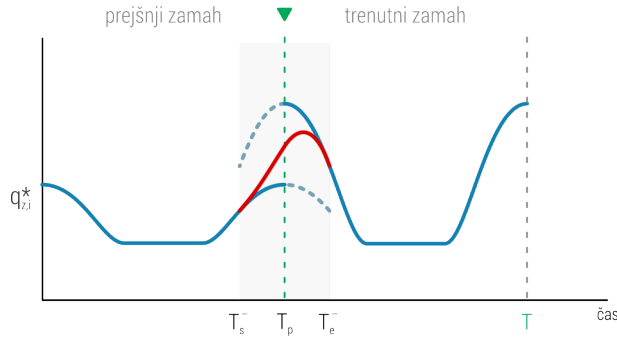
telesu navzdol. Obe omenjeni poenostavitvi naj ne bi bili potrebni, a je brez njih skelet modela lahko po nepotrebnem kompleksen.

Glajenje sosednjih zamahov

Glajenje sosednjih zamahov se računa s pomočjo enačbe, opisane s strani Lee et al. [34]. Funkcija za glajenje dveh krivulj (v tem primeru funkcij zamahov) je polinom šeste stopnje, s katerim se prepreči skok v pospešku spreminjanja krivulje, in zadostuje pogojem:

$$b_{acc}(t_0) = 0, \quad b_{acc}(t_1) = 1, \quad \dot{b}_{acc}(t_i) = 0, \quad \ddot{b}_{acc}(t_i) = 0, \quad \text{za } i = 0, 1, \quad (4.11)$$

s katero se izvaja glajenje na predhodno polnjenimi \mathbf{q}_z^* spremembami sklepov, kot je prikazano na sliki 4.5. Glajenje med sosednjima zamahoma se izvaja 10% najkrajšega časa trajanja prejšnjega zamaha, T^- , in trenutnega zamaha, T , kar je $\min(T^-, T) \frac{1}{10}$. Prednost tega je, da se fizikalen skelet z uporabo nestabilnega PD krmilnika krmili s čim bolj zveznimi spremembami v sklepih.

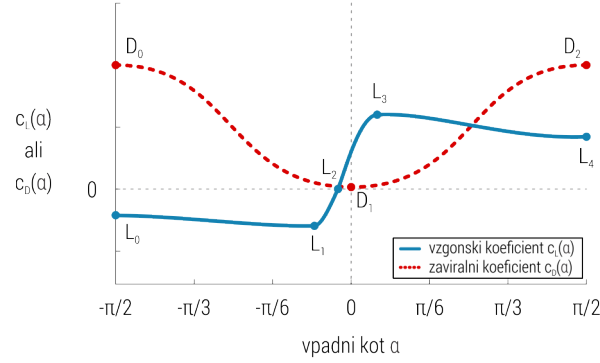


Slika 4.5 Graf prikazuje glajenje prejšnjega zamaha in trenutnega zamaha. Pri tem T_s^- predstavlja začetek glajenja, T_e^- konec glajenja in T^- trajanje prejšnjega zamaha. Siva cona predstavlja regijo glajenja sosednjih zamahov, ki je 10% trajanja najhitrejšega zamaha.

4.1.2 Aerodinamika

Wu & Popović [22] za izračun aerodinamičnih sil uporabljata pristop, predstavljen v 3. poglavju. Ker model kot tak modelira vsako pero posebej in ker osnovne enačbe ne upoštevajo njihovega prekrivanja, Wu & Popović [22] v ta namen za izračun površine segmenta, ki nastopa v enačbi (3.5) uporabita

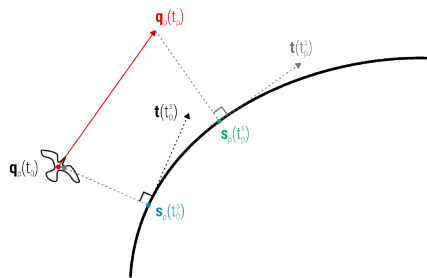
$$S = \lambda_i s_k, \quad (4.12)$$



Slika 4.6 Graf prikazuje funkciji za izračun koeficientov vzgonske sile $C_l(\alpha)$ in zaviralne sile $C_d(\alpha)$ z uporabo karakterističnih točk odvisnih od tipa ptice (x-os radiani, y-os vrednost).

kjer je s_k površina k -tega segmenta obravnavanega peresa in $\lambda_i \in [0, 1]$ korekcijska konstanta površine podskupine peres i . Korekcijska konstanta je izračunana z enačbo $\lambda_i = s_i / \sum_j s_{i,j}$, kjer je s_i površina podskupine peres (primarnih ali sekundarnih), prikazana na sliki 4.3 in $s_{i,j}$ površina peresa j , podskupine peres i . Z željo po skalabilnosti in primerjanju dinamike leta med različnimi tipi ptic sta koeficienta vzgonske in zaviralne sile definirana kot dve zvezni funkciji kosinusih zlepkov (Slika 4.6). Generiranje raznolikih funkcij je zagotovljeno s spreminjanjem petih kontrolnih točk, $L_0 - L_4$, za vzgonski in treh, $D_0 - D_2$, za zaviralni koeficient (Tabela 4.3). Pristnost slednjih, je s strani Wu & Popović [22] zagotovljena zgolj s preizkušanjem in primerjanjem posnetkov gibanj ptic v naravi z generirano animacijo letenja.

4.1.3 Sledenje poti in izračun želene orientacije



Slika 4.7 Prikaz računanja najbližje točke na krivulji $s_p(t_0^s)$ do pozicije ptice $q_p(t_0)$. Rdeča puščica pomeni predikcijo v smeri letenja, s katero se na enak postopek ugotovi predikcijsko točko $s_p(t_p^s)$ na krivulji.

Wu & Popović [22] sta model zasnovala tako, da se optimizira let ptice po vnaprej začrtani poti, ki je predstavljena kot zlepek parametričnih krivulj, s_p . S tem postane po-

	Orel*		Vrana*		Golob		Vrabc*	
vzorčenje [Hz]	1000		1500		2400		4000	
masa [kg]	3.7		0.5		0.25		0.025	
	min	max	min	max	min	max	min	max
F [Hz]	2	5	8	10	8	20	24	30
hitrost [m/s]	0	111	0	60	0	50	0	11
	od	do	od	do	od	do	od	do
k_1	20	1000	2	100	4	40	.01	0.5
k_2	0.02	2	0.002	0.02	1e-4	0.01	1e-5	0.001
L_0	$(-\frac{\pi}{2}, -0.5)$		$(-\frac{\pi}{2}, -0.5)$		$(-\frac{\pi}{2}, -0.1)$		$(-\frac{\pi}{2}, -1)$	
L_1	$(-\frac{\pi}{18}, -1.26)$		$(-\frac{\pi}{18}, -0.8)$		$(-\frac{\pi}{12}, -1.5)$		$(-\frac{\pi}{12}, -2.4)$	
L_2	$(-\frac{\pi}{36}, 0)$		$(-\frac{\pi}{36}, 0)$		$(-\frac{\pi}{36}, 0)$		$(-\frac{\pi}{36}, 0)$	
L_3	$(\frac{\pi}{12}, 1.8)$		$(\frac{\pi}{12}, 1.6)$		$(\frac{\pi}{10}, 5)$		$(\frac{\pi}{12}, 3)$	
L_4	$(\frac{\pi}{2}, 0.5)$		$(\frac{\pi}{2}, 0.5)$		$(\frac{\pi}{2}, 2.2)$		$(\frac{\pi}{2}, 1)$	
D_0	$(-\frac{\pi}{2}, 2)$		$(-\frac{\pi}{2}, 2)$		$(-\frac{\pi}{2}, 1)$		$(-\frac{\pi}{2}, 1)$	
D_1	$(0, 0.02)$		$(0, 0.02)$		$(0, 0.01)$		$(0, 0.01)$	
D_2	$(-\frac{\pi}{2}, 2)$		$(-\frac{\pi}{2}, 2)$		$(-\frac{\pi}{2}, 1)$		$(-\frac{\pi}{2}, 1)$	
w_p	1000		1000		10 000		10 000	
w_v	10		10		100		100	
w_r	300		300		300		300	
w_ω	2		2		2		2	
w_u	1		1		1		1	
w_f	100		100		100		100	
γ_ω	.5		.5		.5		.5	
γ_u	.1		.1		.1		.1	

Tabela 4.3 Prikaz parametrov do sedaj že simuliranih ptic in njihove primerjave. *Povzeto po Wu & Popović [22].

membrno hitro in robustno pridobivanje želene orientacije, \mathbf{q}_r^* , ter pozicije, \mathbf{q}_p^* , tekom leta (Slika 4.7). Na voljo je veliko algoritmov, ki uporabljajo pristope surove sile (angl. *brute-force*), a ob cilju pridobitve dovolj natančnega rezultata se zaradi tega simuliranje lahko zelo upočasnijo. V ta namen je bil s strani Wang, et al. [35] predlagan robusten in učinkovit način pridobivanja najbližje točke na krivulji. Ključne predpostavke sledenja poti so, da se objekt ne približuje krivulji, ampak da v času celotnega poteka simulacije sledi poti ter da ob začetku vemo, kje točno na krivulji se objekt nahaja. Zaradi teh predpostavk je za problem sledenja poti to zadovoljiv algoritem. Problem se lahko pojavi le pri računanju najbližje točke v prihodnosti, kar potrebujemo v koraku ocenjevanja kvalitete zamaha. Ob uporabi istega postopka za napoved pozicije v prihodnosti lahko namreč pride do nezaželenega zapuščanja in vračanja v območje krivulje.

Eden od problemov sledenja poti je pravilno pridobivanje želene orientacije le s pomočjo podane krivulje, \mathbf{s}_p . V ta namen se veliko animacij poslužuje konstrukcije dodatnih krivulj, kar v skalabilni simulaciji letenja ne pride v poštev. Zato se v takih primerih orientacijo točke na krivulji pridobi z uporabo Fernet-Serretovih enačb. V diferencialni geometriji Fernet-Serretove enačbe opisujejo kinematične lastnosti premikajočega delca po neprekinjeni diferencialni krivulji v treh dimenzijah evklidskega prostora \mathbb{R}^3 , oziroma geometrične lastnosti krivulje, neodvisne od katerega koli gibanja. Bolj natančno, enačbe opisujejo odvode, imenovane tangentni, normalni in binormalni enotski vektor, ki so v odvisnostih med seboj.

Tangentni, normalni in binormalni enotski vektorji, ki skupaj tvorijo ortogonalno bazo v \mathbb{R}^3 , so definirani z enačbami

$$\begin{aligned} \mathbf{t}(t) &= \frac{\dot{\mathbf{s}}_p(t)}{\|\dot{\mathbf{s}}_p(t)\|} \\ \mathbf{b}(t) &= \frac{\dot{\mathbf{s}}_p(t) \times \ddot{\mathbf{s}}_p(t)}{\|\dot{\mathbf{s}}_p(t) \times \ddot{\mathbf{s}}_p(t)\|} \\ \mathbf{m}(t) &= \mathbf{t}(t) \times \mathbf{b}(t), \end{aligned} \tag{4.13}$$

kjer je t parameter krivulje, $\dot{\mathbf{s}}_p$ prvi odvod krivulje po parametru in $\ddot{\mathbf{s}}_p$ drugi odvod krivulje po parametru. Končni rezultat teh enačb je koordinatni sistem, ki se zvezno spreminja in vrta v odvisnosti od poteka krivulje.

Za odpravo ostrih preskokov koordinatnega sistema je v primeru zlepkov treba uporabiti vsaj krivulje četrtega reda. Pri višjih redih krivulja postane parametrično ne-intuitivna in veliko bolj neobvladljiva v smislu določanja poti leta. Za parametrično

krivuljo se ukrivljenost izračuna z enačbo

$$\begin{aligned}\kappa(t) &= \frac{1}{R(t)} \\ \kappa(t) &= \frac{\|\dot{\mathbf{s}}_p(t) \times \ddot{\mathbf{s}}_p(t)\|}{\|\dot{\mathbf{s}}_p(t)\|^3},\end{aligned}\tag{4.14}$$

kjer je κ ukrivljenost krivulje in R radij ukrivljenosti krivulje.

Ne glede na izbiro algoritma za računanje najbližje točke, je mogoče iz krivulje dobiti približek želene orientacije. Naj bo t_0 začetek in t_1 konec zamaha ter naj bosta t_0^s in t_1^s pripadajoči vrednosti parametra krivulje tako, da je $\mathbf{s}_p(t_i^s)$ točka na krivulji najbližje $\mathbf{q}(t_i)$. Enačbe

$$\begin{aligned}\mathbf{a}_c &= \kappa(t_1^s) \|\dot{\mathbf{q}}_p(t_1)\|^2 \mathbf{m}(t_1^s) \\ \mathbf{a}_t &= \frac{-\max(0, \|\dot{\mathbf{q}}_p(t_0)\| - \|\dot{\mathbf{q}}_p(t_1)\|)}{\Delta t} \mathbf{t}(t_1^s) \\ \mathbf{a}^* &= \mathbf{a}_c + \mathbf{a}_t - \mathbf{g},\end{aligned}\tag{4.15}$$

so zastavljene s predpostavko, da ptica z zamahovanjem generira pospešek v smeri navzgor, relativno na orientacijo trupa, kjer je \mathbf{a}_c centrifugalni pospešek, \mathbf{a}_t tangenta zaviralna sila in \mathbf{a}^* željen pospešek. Ker se s tem pri učenju zamahov zagotovi le gibanje vzporedno poti, je treba definirati spajalno krivuljo, s katero orientacijo usmerimo proti originalni poti leta. To pomeni, da se namesto $\mathbf{t}(t_1^s)$ uporabi tangenti enotski vektor $\mathbf{t}^+(t_1^s)$ spajalne krivulje, \mathbf{s}_p^+ . Enostavneje se ta izračuna s

$$\mathbf{t}^+(t_1^s) = \frac{\mathbf{s}_p(t_p^s) - \mathbf{q}_p(t_1)}{\|\mathbf{s}_p(t_p^s) - \mathbf{q}_p(t_1)\|},\tag{4.16}$$

kjer je $\mathbf{s}_p(t_p^s)$ napovedana točka na krivulji in $\mathbf{q}_p(t_1)$ pozicija, obe na koncu enega zamaha. Želena orientacija na podlagi krivulje se tako izračuna z enačbami

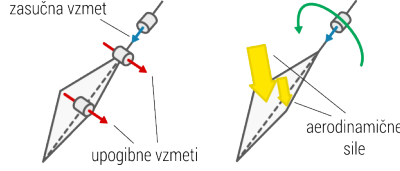
$$\begin{aligned}\mathbf{R}^* &= \begin{bmatrix} \mathbf{r}_x & \mathbf{r}_y & \mathbf{r}_z \end{bmatrix}, \\ \mathbf{r}_x &= \mathbf{r}_y \times \mathbf{t}^+(t_1^s), \quad \mathbf{r}_y = \frac{\mathbf{a}^*}{\|\mathbf{a}^*\|}, \quad \mathbf{r}_z = \mathbf{r}_x \times \mathbf{r}_y\end{aligned}\tag{4.17}$$

in pretvorbo matrike \mathbf{R}^* v kvaternion $\mathbf{q}_r^*(t_1)$.

Deformacija peres

Zelo na kratko je omenjena tudi uporaba upogibanja in zvijanja posameznih peres (Slika 4.8). Wu & Popović [22] tega sicer nikjer ne omenjata, a naše izkušnje kažejo, da je modeliranje ukrivljanja in zvijanja ključno pri ustvarjanju dovolj močne vzgonske sile pri

stranskem zamahovanju kril, s katerim je ptici omogočena možnost spreminjanja smeri letenja. Nekateri modeli, kot je [23], možnost spreminjanja smeri letenja ptice rešuje s korekcijo skeleta, kjer v zapestnem delu poleg iztega dovoljujejo tudi zamahovanje.



Slika 4.8 Prikaz podrobnejše konstrukcije peres. LEVO upogibanje peresa, DESNO zvijanje peresa okoli svoje osi (*pritisk zraka lahko generira različne sile na sosednjih straneh peresa*).

4.1.4 Optimizacija parametriziranih zamahov

Z željo po izdelavi modela, zmožnega učenja zamahov, je treba dinamiko generiranega zamaha tudi primerno oceniti. Predstavljen učni model je za zagotavljanje učenja optimalnih zamahov razdeljen na minimizacijo odstopanja končne pozicije, hitrosti in orientacije, kar sili učni model k čim boljšem sledenju predhodno definirani poti. Ker sledenje poti ni dovolj dobro zagotovilo generiranja verodostojnih zamahov, sta Wu & Popović [22] v času učenja model silila k minimizaciji nenadnih sprememb orientacije trupa in k čim manjši porabi energije ob zamahovanju navzgor. Modeliranje upogibanja in zvijanja peres pa je zahtevalo tudi ocenjevanje premikanja kril proti smeri letenja.

S tem namenom je treba v učnem modelu hraniti čas začetka zamaha, t_0 , in čas konca zamaha, t_1 , s celotnim trajanjem slednjega, $\Delta t = t_1 - t_0$. Ocenjevanje letenja s pomočjo predhodno definirane poti zahteva tudi ločeno vodenje časov krivulje oziroma vrednosti parametra ob začetku zamaha, t_0^s , in koncu zamaha, t_1^s . Ocenjevanje premagovanja poti se razdeli na dva dela, pri čemer je prvi del namenjen le ocenjevanju zmožnost premagovanja poti

$$\begin{aligned} E_p &= \|\mathbf{q}_p(t_1) - \mathbf{s}_p(t_1^s)\|^2 \\ E_v &= \|\dot{\mathbf{q}}_p(t_1) - \mathbf{v}(t_1^s)\|^2 \\ E_r &= \psi^2(\mathbf{q}_r(t_1), \mathbf{q}_r^*(t_1)), \end{aligned} \quad (4.18)$$

s katerimi se po zaključku zamaha oceni oddaljenost od poti, E_p , hitrost letenja, E_v , pri čemer je treba $\mathbf{v}(t_1^s)$ željeno hitrost poti voditi ločeno, in orientacijo, E_r , kjer je ψ najmanjši pozitiven kot med dvema kvarternionoma, ki se najlažje izračuna kot

$$\psi(\mathbf{q}_r^0, \mathbf{q}_r^1) = \arccos(2(\mathbf{q}_r^0 \cdot \mathbf{q}_r^1)^2 - 1), \quad (4.19)$$

kjer sta \mathbf{q}_r^0 in \mathbf{q}_r^1 v tem vrstnem redu začetna in končna orientacija predstavljeni kot kvarterniona.

Prva enačba drugega dela je namenjena kaznovanju ne-gladkega spreminjanja orientacije, z nepotrebno uporabo prevelike rotacijske hitrosti

$$E_\omega = \max \left(0, \gamma_\omega \int \|\omega\| dt + (1 - \gamma_\omega) \max_t (\|\omega\|) \Delta t - \psi(\mathbf{q}_r(t_0), \mathbf{q}_r^*(t_1)) \right)^2 \quad (4.20)$$

s katero se zagotovi stabilnejši let v času celotnega zamaha, kjer je ω rotacijska hitrost trupa. Integracijski del predstavlja vsoto sprememb rotacijskih hitrosti in funkcija maksimuma hrani najbolj intenzivno pojavitev v času zamaha. Konstanta γ_ω omogoča nastavljanje pomembnosti med integralnim in maksimalnim delom rotacijske hitrosti.

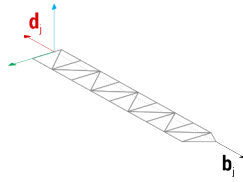
Naslednja enačba je namenjena temu, da se v času zamaha prepreči prekomerno gibanje sklepov v smeri navzgor. Osnovni koncept zamaha dol je, da ptica s tem ohranja višino leta, pri čemer ima ptica v času zamaha gor veliko več različnih možnosti gibanja

$$E_u = \frac{1}{m^2} \sum_{i \in \mathcal{V}} \left(\gamma_u \int \tau_i^2 \nu(t, u_\alpha) dt + (1 - \gamma_u) \max_t (\tau_i^2 \nu(t, u_\alpha)) \right), \quad (4.21)$$

kjer je \mathcal{V} množica obeh sklepov, namenjenih zvijanju, m masa ptice in $\nu(t, u_\alpha)$ funkcija, s katero se pridobi stanje zamaha. V modelih z upogibajočimi peresi je prav tako treba kaznovati gibanje peres v smeri letenja. Ker bi tako gibanje vznemirilo in po vsej verjetnosti poškodovalo peresa prave ptice se za preprečevanje slednjega uporabi enačbo

$$E_f = \frac{1}{\Delta t} \int \max_{j \in \mathcal{F}} (b_j^2(t)) dt, \quad (4.22)$$

ki kaznuje gibanje konic peres proti smeri letenja $b_j(t) = \max(0, \mathbf{d}_j \cdot \mathbf{v})$, kjer je \mathcal{F} množica vseh peres, namenjenih letenju, \mathbf{d}_j orientacija peresa (Slika 4.9) in \mathbf{v} hitrost zraka, relativna na krilo.



Slika 4.9 Prikaz \mathbf{d}_j orientacije peresa in \mathbf{b}_j smeri kaznovanja gibanja.

Ocenjevalna funkcija je v večjih primerih vsota vseh skupaj, kjer se z željo po kontroliranju pomembnosti vsake posebej sestavi obtežena vsota

$$E = w_p E_p + w_v E_v + w_r E_r + w_\omega E_\omega + w_u E_u + w_f E_f. \quad (4.23)$$

Učenje zamahov z uporabo simuliranega ohlajanja

Učenje zamahov z metodo simuliranega ohlajanja je zaradi prevelike stopnje preiskovalnega prostora iz čisto naključnega stanja nemogoče. Zato je treba vnaprej nastaviti zamah letenja, s katerim se ptica čim bolj stabilno giblje, poravnano glede na tla. Tak koncept učenja z „začetnim zamahom“ je v osnovi dovolj dober za optimizacijo gibanja po želeni poti. A ker je za model, ki naj bi bil zmožen krmiljenja po prostoru, treba hraniti čim več naučenih zamahov, se naj bi „začetni zamah“ uporabil le ob prvi optimizaciji, za vsako naslednjo bi se izbral naslednji najbolj primeren iz množice že naučenih zamahov, s katerim bi potencialno lahko dosegli želeni cilj. Na ta način se iz že enkrat optimiziranega stanja lahko po optimizaciji dobi novo, boljše stanje.

Algoritem simuliranega ohlajanja je za tovrsten primer nastavljen tako, da se zanka ohlajanja omeji na maksimalno tisoč iteracij. Nastavljena temperatura 10°C , pri čemer je treba biti pozoren, da je zelo odvisna od kvalitete „začetnega zamaha“. To pomeni, da če je ocena zamaha manjša od nastavljene temperature, se s tem zagotovi bolj razpršeno preiskovanje prostora (kljub ročni omejitvi iteracije algoritma se nastavi minimalno temperaturo 0.01°C). Ohlajanje algoritma se izvede vsakih sto iteracij za 10%. Iz prejšnjega stanja se izvede 10% mutacija enega naključno izbranega parametra vektorja stanja \mathbf{u} .

4.2 Poenostavljen model

Z razliko od Wu & Popović [22], ki sta si za cilj zadala čim bolj verodostojno gibanje, namenjeno filmski industriji, ne ozirajoč se na čas simulacije, so si Zhu, et al. [23] za cilj zadali model, zmožen animiranja zamahov v realnem času (predstavljen z minimalnim naborom parametrov, vidnih v tabeli 4.4). Model je, za razliko od večine predstavljenih, mogoče krmiliti skozi predhodno nastavljene točke oziroma slediti predhodno nastavljeni poti v realnem času. Dinamika letenja ptice se računa z uporabo eksplicitne Eulerjeve integracije:

$$\ddot{\mathbf{q}}_p(t) = \frac{\mathbf{F}(t)}{m} \quad (4.24)$$

$$\dot{\mathbf{q}}_p(t + \Delta t) = \dot{\mathbf{q}}_p(t) + \ddot{\mathbf{q}}_p(t)\Delta t, \quad (4.25)$$

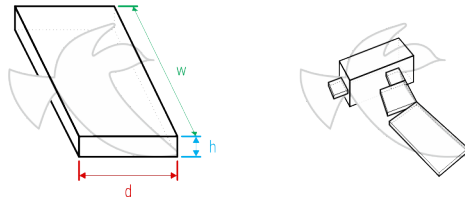
$$\mathbf{q}_p(t + \Delta t) = \mathbf{q}_p(t) + \dot{\mathbf{q}}_p(t)\Delta t,$$

kjer je \mathbf{F} sila, ki deluje na ptico (vsota gravitacijske sile, \mathbf{F}_g , in ostalih aerodinamičnih sil), in m masa ptice. Sila in masa določata pospešek, $\ddot{\mathbf{q}}_p(t)$, s čemer se izračuna novo

	Orel			Škorec		
masa [g]	2000			500		
S [cm ²]	7200			1760		
B [cm]	90			40		
l_t [cm]	40			20		
	w	h	d	w	h	d
dimenzija [cm]	220	20	80	100	10	40
	min		max	min		max
F [Hz]	2		5	8		10
A	0		1	0.6		1
$q_{t,p}$ [°]	-10		10	-10		10
$q_{t,r}$ [°]	-45		45	-60		60
$q_{t,s}$ [°]	10			10		
γ_F	0.36			1.0		
γ_A	0.25			0.25		
γ_t	1.78			1.78		
k	10			10		

Tabela 4.4 Prikaz simulacijskih parametrov povzetih po Zhu, et al. [23].

hitrost, $\dot{\mathbf{q}}_p(t + \Delta t)$, in pozicijo, $\mathbf{q}_p(t + \Delta t)$. Za razliko Wu & Popović [22], ki ptico modelirata izredno podrobno, jo Zhu, et al. [23] modelirajo močno poenostavljeno. V njihovem primeru ima krilo zgolj dva sklepa, in sicer ramenski in zapestni, oba s tremi prostostnimi stopnjami. Krilo ločijo zgolj na dva dela, notranji in zunanji, ki ga tudi ne modelirajo kot sklop peres. Predstavljeni model krila uporablja striktno le za regulacijo spreminjanja višine leta in rep kot primarni regulator spreminjanja orientacije celotne ptice. Trup modela je definiran kot enostaven kvader (Slika 4.10), na katerem so pritrjena krila in rep, ki se obravnavajo kot brezmasni objekti.



Slika 4.10 Prikaz označevanja dimenzij členov ptice. LEVO zelo poenostavljeno fizikalno telo ptice Zhu, et al. [23], DESNO podrobnejša konstrukcija polnofunkcionalnega fizikalnega telesa, Wu & Popović [22].

4.2.1 Reprezentacija zamaha

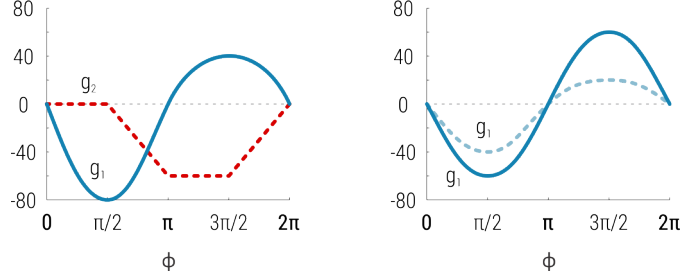
Predstavljeni model, za razliko od modela Wu & Popović [22], za spreminjanje sklepov uporablja kinematiko naprej (angl. *forward kinematics*). To pomeni, da model ne potrebuje parametriziranega stanja zamaha, \mathbf{u} , ampak samo kote sklepov, za namen verižnega apliciranja le teh od korenskega sklepa naprej. Zhu, et al. [23] ne predstavijo načina računanja zveznega spreminjanja kotov v sklepih, prikažejo zgolj slike. Funkciji g_1 in g_2 , ki določata spreminjanje zamahovanja in uklanjanja ramenskega in zapestnega sklepa, sta definirani, kot prikazuje slika 4.11. Za spreminjanje zvijanja, tako v ramenskem kot v zapestnem sklepu, ne podajo nobene informacije.

Za lažjo primerljivost med modeli, se iz predloženih shem (Slika 4.11) lahko rekonstruira funkciji zamahovanja. Ti funkciji se v tem primeru predstavi primerljivo modelu Wu & Popović [22], z opcijo ročnega nastavljanja maksimalne in minimalne meje gibanja sklepa. Za rekonstrukcijo take funkcije je treba definirati zlepek $s(x, y_0, y_1)$ kot polinom pete stopnje, ki zadostuje pogojem:

$$s\left(-\frac{1}{2}\right) = \underline{q}_{z,i}, \quad s(0) = 0, \quad s\left(\frac{1}{2}\right) = \bar{q}_{z,i}, \quad \dot{s}\left(\frac{1}{2}\right) = \dot{s}\left(-\frac{1}{2}\right) = -1, \quad (4.26)$$

kjer sta $\underline{q}_{z,i}$ in $\bar{q}_{z,i}$ v tem vrstnem redu minimalna in maksimalna vrednost funkcije,

definirane na intervalu $[-\frac{1}{2}, \frac{1}{2}]$, ki vedno seka x -os. Za zagotovitev sekanja x -osi in preprečitev neželene oscilacije na začetku in koncu zlepk, je dovolj, če se parametra $\underline{q}_{z,i}$ in $\bar{q}_{z,i}$ izbereta v odvisnosti $-2\bar{q}_{z,i} \leq \underline{q}_{z,i} \leq -\bar{q}_{z,i}$.



Slika 4.11 Prikaz funkcij računanja kotov g_1 in g_2 (x -os radiani, y -os stopinje). Krivulje z enojno črta se uporabljajo za roko in črtkana črta za dlani. LEVO koti zamahovanja, DESNO koti zvijanja.

Funkcija g_1 se definira

$$g_1(\underline{q}_{z,i}, \bar{q}_{z,i}, \phi) = \begin{cases} s(\frac{\phi}{\pi} - \lfloor \frac{1}{2} + \frac{\phi}{\pi} \rfloor, \underline{q}_{z,i}, \bar{q}_{z,i}) & \lfloor \frac{1}{2} + \frac{\phi}{\pi} \rfloor \equiv 0 \pmod{2} \\ s(\frac{\phi}{\pi} - \lfloor \frac{1}{2} + \frac{\phi}{\pi} \rfloor, \bar{q}_{z,i}, \underline{q}_{z,i}) & \text{ostalo} \end{cases}, \quad (4.27)$$

ki v omenjenem modelu predstavlja način gibanja roke pri zamahovanju in zvijanju ter gibanje dlani pri zvijanju. Funkcija g_2 je po drugi strani le skupek linearnih zlepkov, ki predstavljajo gibanje dlani pri zamahovanju. Reguliranje frekvence zamahovanja določa enačba

$$F(t + \Delta t) = \begin{cases} \bar{F} & F(t) > \bar{F} \\ F(t) + \gamma_F \text{sgn}(T_y - \mathbf{q}_{p,y}) \Delta t & \underline{F} \leq F(t) \leq \bar{F} \\ \underline{F} & F(t) < \underline{F} \end{cases}, \quad (4.28)$$

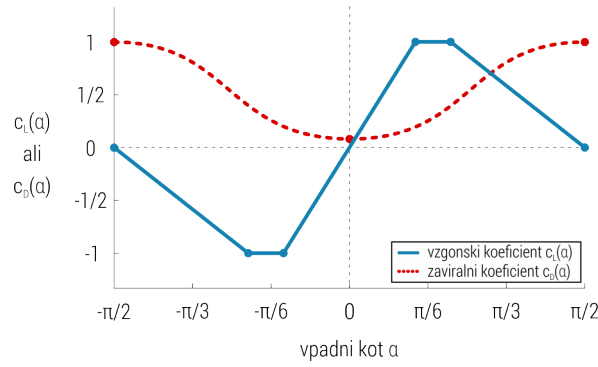
amplitudo zamahovanja pa

$$A(t + \Delta t) = \begin{cases} \bar{A} & A(t) > \bar{A} \\ A(t) + \gamma_A \text{sgn}(T_y - \mathbf{q}_{p,y}) \Delta t & \underline{A} \leq A(t) \leq \bar{A} \\ \underline{A} & A(t) < \underline{A} \end{cases}, \quad (4.29)$$

kjer je F frekvenca zamaha (Hz), $A \in [0, 1]$ koeficient reguliranja amplitude zamaha, T_y y -komponenta ciljne točke (Slika 4.14), $\mathbf{q}_{p,y}$ y -komponenta trenutne pozicije ptice, $\gamma_F \in [0, 1]$ koeficient hitrosti spreminjanja frekvence in $\gamma_A \in [0, 1]$ koeficient hitrosti spreminjanja amplitude.

4.2.2 Aerodinamika

Zaradi želje po čim hitrejšem izvajanju simulacije se aerodinamične sile, ki delujejo na krilo računajo zgolj v dveh točkah, središču notranjega dela krila in središču zunanega dela krila. Funkciji spreminjanja koeficientov vzgona in upora (Slika 4.12) so izmišljene in nimajo nobene podlage. Obe funkciji za razliko od Wu & Popović [22], kjer regulirata potek s kontrolnimi točkami, Zhu, et al. [23] definirajo na fiksnem intervalu $[0, 1]$, pri čemer je izračun vzgonskega koeficienta močno poenostavljen z uporabo linearnih zlepkov, s čimer zanemarijo zvezno spreminjanje slednjega pod različnimi vpadnimi koti.



Slika 4.12 Graf prikazuje funkciji za izračun koeficientov vzgonske sile $C_l(\alpha)$ in zaviralne sile $C_d(\alpha)$ (x-os radiani, y-os faktor $[-1, 1]$).

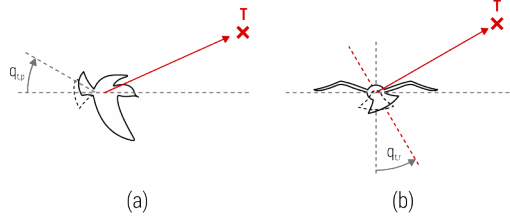
Pri enačbi za izračun aerodinamičnih sil, enačba (3.3), se površina repa izračuna z enačbo

$$S = \frac{l_t^2 q_{t,s}}{2}. \quad (4.30)$$

Ker je rep skonstruiran kot pahljača trikotnikov, sta za izračun potrebna dolžina repnega peresa, l_t , in kot odprtosti repa v radianih, $q_{t,s}$. Kot odprtosti se spreminja s predpostavko, da se površina repa spreminja zgolj v odvisnosti od zvijanja in upogibanja repa, in se izračuna z enačbo

$$q_{t,s} = \gamma_t |\max(q_{t,p}, q_{t,r})| + \underline{q}_{t,s}, \quad (4.31)$$

kjer je $\gamma_t \in [0, 1]$ koeficient regulacije odprtosti repa, $\underline{q}_{t,s}$ minimalni kot odprtosti repa v radianih, $q_{t,p}$ in $q_{t,r}$ pa sta po vrsti upogib in naklon repa, kot prikazuje slika 4.13. Iz te poenostavitve sledi, da je rep ptice v času naravnega položaja v zaprtem stanju, oziroma se upošteva le $\underline{q}_{t,s}$, kar pri ravnem letu prepreči apliciranje vzgonskih in zaviralnih sil. Ob spreminjanju smeri se $q_{t,s}$ kot odprtosti začne povečevati, kar privede do spreminjanja orientacije celotnega trupa in posledično tudi smeri letenja.



Slika 4.13 Prikaz računanja kotov upogiba, $q_{t,p}$, in zvijanja repa, $q_{t,r}$, v odvisnosti od ciljne točke T in orientacije ptice: (a) dviganje-spuščanje (b) zavijanje/nagibanje levo-desno.

S pomočjo ciljne točke letenja, T , se določi tudi kotna hitrost za smer letenja ptice. Dviganje in spuščanje, $q_{r,p}^*$, ter zvijanje levo-desno, $q_{r,y}^*$, se preprosto postavi tako, da vektor smeri letenja sovpada s smerjo poti ciljni točki, čemur se prilagodi tudi nagib, $q_{r,r}^*$ (Slika 4.14). Trenutna kotna hitrost se izračuna z uporabo enačbe

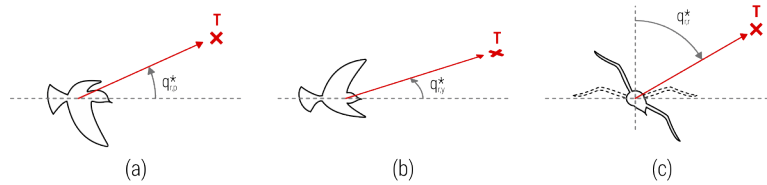
$$\boldsymbol{\omega}(t) = -\text{sgn}(q_{r,r}^*)\mathbf{r} \quad (4.32)$$

kjer je \mathbf{r} standardna kotna hitrost. Na podlagi vsote aerodinamičnih sil generiranih z repom, $\mathbf{F}_t(t)$, se izračuna kotni pospešek telesa, definiran z enačbami

$$\begin{aligned} \boldsymbol{\tau}(t) &= \mathbf{x} \times \mathbf{F}_t(t) \\ \boldsymbol{\beta}(t) &= \mathbf{I}^{-1} \cdot \boldsymbol{\tau}(t), \end{aligned} \quad (4.33)$$

kjer je $\boldsymbol{\tau}(t)$ navor, \mathbf{x} momentni vektor od centra trupa do točke izračuna sile $\mathbf{F}_t(t)$ in $\mathbf{F}_t(t)$ vsota aerodinamičnih sil repa. Za pravilen izračun je potrebno le nastaviti \mathbf{I} , vztrajnostni moment telesa, katerega se za enostaven 3D kvader (namenjen vrtenju okoli svojega središča) izračuna kot

$$\mathbf{I} = \frac{m}{12} \begin{bmatrix} h^2 + w^2 & 0 & 0 \\ 0 & w^2 + d^2 & 0 \\ 0 & 0 & d^2 + h^2 \end{bmatrix}, \quad (4.34)$$



Slika 4.14 Prikaz računanja kotov v odvisnosti od ciljne točke, T , in orientacije ptice: (a) dviganje-spuščanje (angl. *pitch*), $q_{r,p}^*$, (b) zavijanje levo-desno (angl. *yaw*), $q_{r,y}^*$, (c) nagibanje levo-desno (angl. *roll*), $q_{r,r}^*$.

kjer je m masa celotne ptice. Za spremembo orientacije trupa je, kot pri premikanju po prostoru, uporabljena eksplicitna Eulerjeva integracija

$$\boldsymbol{\omega}(t + \Delta t) = \boldsymbol{\omega}(t) + \boldsymbol{\beta}(t)\Delta t, \quad (4.35)$$

s katero se izračuna posodobljen rotacijski pospešek. Za računanje spremembe orientacije je v fizikalnih simulacijah bolje računati s kvarternioni

$$\mathbf{q}_r(t + \Delta t) = \mathbf{q}_r(t) + \frac{1}{2} \begin{bmatrix} 0 \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \otimes \mathbf{q}_r(t)\Delta t. \quad (4.36)$$

4.2.3 Vpliv tal

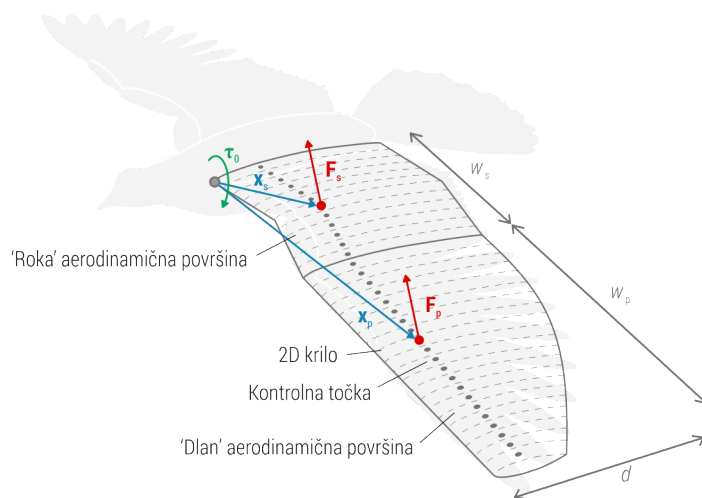
Celovit model letenja ptice, ki je zmožen manevriranja, mora imeti tudi možnost vzleta iz stanja mirovanja ali samo jadranje čisto pri tleh. V ta namen so Zhu, et al. [23] dodali vpliv tal. Podaja ga enačba

$$\mathbf{F}_{ground}(h) = \begin{cases} k \frac{-\mathbf{F}_g m}{h}, & 0 < \frac{h}{B} < 1 \\ 0 & \text{ostalo} \end{cases}, \quad (4.37)$$

kjer je k koeficient vpliva tal, m masa ptice, h višina letenja, \mathbf{F}_g gravitacijska sila in B razpon krila ptice.

4.3 Model z upoštevanjem zračnih tokov

Ob predstavitvi veliko modelov, ki uporabljajo dinamiko naprej [22] ali kinematiko naprej [23], so Parslew [24, 36] in Parslew & Crowther [37] po drugi strani predstavili model, ki za spreminjanje sklepov modela uporablja dinamiko nazaj (angl. *inverse dynamics*). Namen celotnega modela je le generiranje zamahov pod različnimi nakloni ter hitrostmi leta, ki pa ni namenjen interaktivnemu krmiljenju v realnem času. Model za generiranje raznolikih zamahov izvaja optimizacijo kotov v sklepih, z upoštevanjem zračnih tokov in porabe energije za njihovo doseg. Krilo ptice je v kompleksnosti najbolj primerljivo modelu Wu & Popović [22]. V modelu je ramenski sklep predstavljen s tremi prostostnimi stopnjami, zapestni sklep z dvema prostostnima stopnjama (uklanjanje naprej-nazaj in zvijanje), komolčni sklep z eno prostostno stopnjo (uklanjanje naprej-nazaj) in rep brez prostostnih stopenj. S predpostavko, da je modeliranje vsakega peresa posamično nepotrebno, se

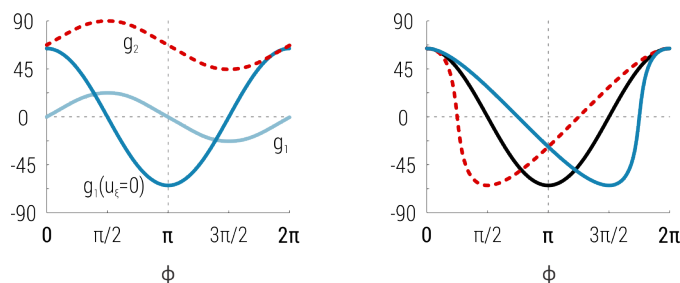


Slika 4.15 Prikaz kontrolnih točk in površine krila (2D kapljičasta geometrija računanja zračnih vrtincev) z dolžino notranjega dela krila w_s , dolžino zunanjega dela krila w_p in širino d .

površina krila razdeli na dve aerodinamični površini, zunanji in notranji del krila, napeti na skelet ptice, kot je to prikazano na sliki 4.15. Vsaka aerodinamična površina je sestavljena iz segmentov pravokotnikov, namenjenih računanju aerodinamičnih sil na podlagi zračnih tokov, z uporabo teorije elementov (angl. *blade-element theory*).

4.3.1 Reprezentacija zamaha

Za razliko od modela Wu & Popović [22] je v tem modelu veliko bolj razdrobljeno opisana struktura \mathbf{u} vektorja parametrov zamaha. Kljub temu se lahko s pomočjo predstavljenih enačb (Slika 4.16) izdela tak vektor, ki je bolj primerljiv modelu [22] (Tabela 4.5). Funkcije za izračun želene pozicije sklepa skeleta so definirane kot



Slika 4.16 LEVO Prikaz funkcij računanja kotov g_1 in g_2 (x-os radiani, y-os stopinje), DESNO prikaz uporabe fazne transformacije Wu & Popović [22] s funkcijo g_1 .

$$\begin{aligned}
g_1(u_i^s, u_i^t, u_\xi, \phi) &= u_i^s \cos(\phi + u_\xi) + u_i^t \\
g_2(u_i^f, u_\varsigma, \phi) &= \frac{1}{2}(1 - u_i^f) \cos(\phi + u_\varsigma) + \frac{1}{2}(1 + u_i^f),
\end{aligned} \tag{4.38}$$

kjer se g_1 z $u_\xi = 0$ uporablja za spreminjanje kotov zamahovanja ter s spremenljivim parametrom u_ξ za kote zvijanja, s katerim dosežemo poljuben zamik faze. Rezultat funkcije g_1 so koti, namenjeni spreminjanju kotov zglobov, iz česar sledi, da sta parametra u_i^s in u_i^t hranjena v radianih ter sta v tem vrstnem sprememba amplitude in kotni zamik. Funkcija $g_2 \in [0, 1]$ je namenjena računanju faktorjev uklanjanja naprej-nazaj, iz česar

Parameter	Stran	Opis
T		Trajanje zamaha
$u_\alpha, u_\xi, u_\varsigma$		Regulacijske spremenljivke zamaha
u_1^s, u_1^t		Rama kot zamahovanja
u_2^s, u_2^t		Rama kot zvijanja
u_3^f		Rama kot uklanjanja naprej-nazaj
u_4^f		Komolec kot uklanjanja naprej-nazaj
u_5^f		Zapestje kot zvijanja
u_6^f		Zapestje kot uklanjanja naprej-nazaj

Tabela 4.5 Prikaz vektorja parametrov zamaha \mathbf{u} . Število vseh parametrov je 12, od tega trije korekcijski in en čas.

sledi, da so parametri $u_i^f \in [0, 1]$. Pretvorba v kote sklepov se izvede z

$$q_{z,i}^*(t) = \underline{q}_{z,i} + (\bar{q}_{z,i} - \underline{q}_{z,i})g_2(u_i^f, u_\varsigma, \phi), \tag{4.39}$$

kjer sta $\underline{q}_{z,i}$ in $\bar{q}_{z,i}$ v tem vrstnem redu maksimalen in minimalen kot sklepa. Z željo po čim manjši kompleksnosti modela ptice, Parslew v model vpelje tudi avtomatiko zapiranja in odpiranja krila, z enakim namenom, kot to predstavita Wu & Popović [22]. Za zagotovitev le tega, se pri zvijanju zapestnega sklepa, namesto omenjene uporabe g_1 , uporabi g_2 posledično z enačbo (4.39). Iz predstavljenega modela se prav tako lahko

	u_α	u_ξ [°]	u_ς [°]	u_1 [°]	u_2 [°]	u_3 [°]	u_4 [°]	u_5 [°]	u_6 [°]
max	.7	90	90	90	75	48	85	0	0
min	.3	0	0	-90	-75	90	180	90	90

Tabela 4.6 Prikaz omejitev sklepov ptice. Koti se obravnavajo relativno na korenski sklep, pri čemer se upošteva, da je ramenski sklep pravokotno poravnal s telesom.

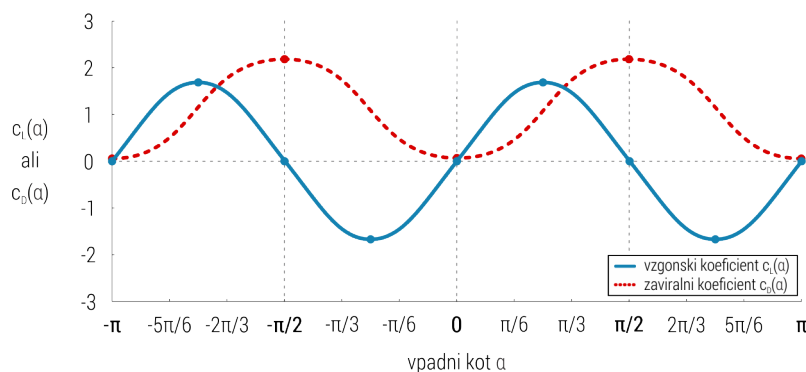
razbere meje sklepov modelirane ptice, prikazane v tabeli 4.6. Edina slabost teh je, da so s strani Parslewa pridobljeni le iz slik gibanja ptic.

4.3.2 Aerodinamika

Aerodinamične sile se v večini predstavljenih modelov računa s poenostavljenimi aerodinamičnimi enačbami, medtem ko jih Parslew računa z upoštevanjem generiranih zračnih tokov, ki nastajajo ob zamahovanju s krili. Z namenom primerjave med modeli bomo tu izpostavili zgolj način določanja koeficientov vzgona in upora, za kar se uporablja enačbi

$$\begin{aligned} C_l(\alpha) &= A \sin(2\alpha) \\ C_d(\alpha) &= B + C \cos(2\alpha). \end{aligned} \quad (4.40)$$

kjer so $A = 1.64$, $B = 1.135$, $C = -1.05$ konstante, ki sovpadajo z meritvami [24]. Parslew med drugim opozori na nenormalno visoke vrednosti vzgonskega koeficienta v območju vpadnih kotov blizu $\frac{\pi}{2}$ v modelu, ki sta ga predstavila Wu & Popović [22].

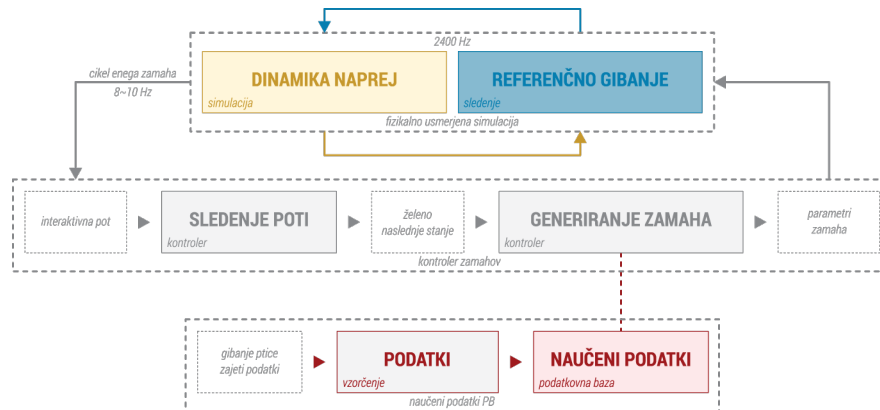


Slika 4.17 Graf prikazuje funkciji za izračun koeficientov vzgonske sile $C_l(\alpha)$ in zaviralne sile $C_d(\alpha)$ s spreminjanjem le treh parametrov.

4.4 Podatkovno usmerjeni model

Velik korak v izdelavi krmilnika, ki bi bil sposoben manevriranja fizikalno zasnovanega modela ptice v realnem času, je bil predstavljen s strani Ju, et al. [25, 38], kjer je osnovni koncept generiranja zamahov iz vnaprej zajete baze referenčnih zamahov in želene smeri letenja. Za izdelavo tovrstnega krmilnika je bil uporabljen pravi golob, katerega prednost pred drugimi pticami je, da se veliko lažje usmerja. Na letalna peresa in trup ptice so pritrdili odsevnike in na podlagi njihove zajete poti, rekonstruirali dinamiko gibanja sklepov skeleta. Podatki so bili zajeti s pomočjo visoko zmogljivih kamer pri 240 sličicah na sekundo v prostoru višine 10m, širine 10m in višine 7m. S tem postopkom pridobivanja učne množice zamahov in uporabe nenadzorovanega učenja so tako pridobili referenčno

gibanje vseh sklepov skeleta ptice. Računalniški model je v smislu kompleksnosti, od vseh do sedaj omenjenih, najbolj primerljiv modelu Wu & Popović [22], ker prav tako upošteva ukrivljanje in zvijanje peres ter računanje vzgonskih in zaviralnih koeficientov sestavljenih iz kubičnih zlepkov.



Slika 4.18 Prikaz kontrolerja zamahovanja s krili, ki pridobiva želeno smer letenja iz predhodno definirane poti in generiranje zamahov iz podatkovne baze.

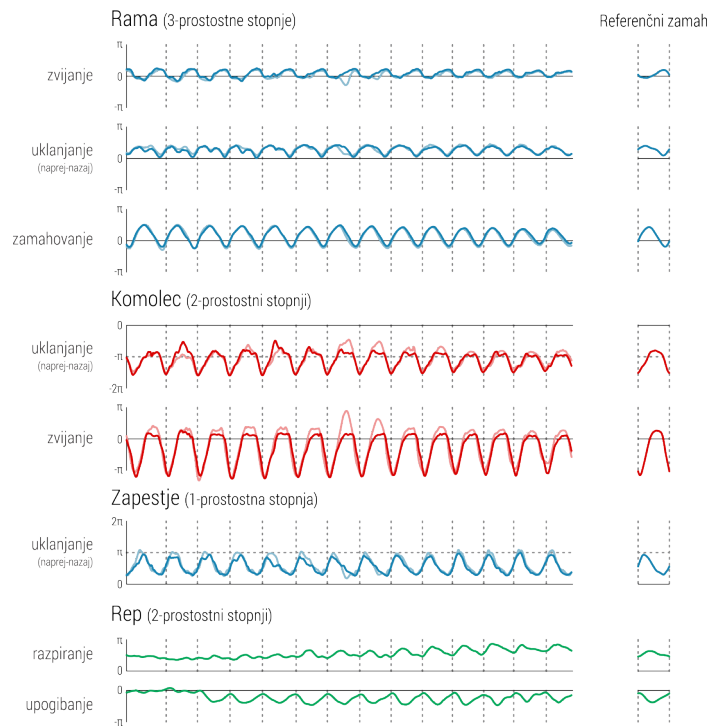
Krmilnik letenja so v osnovi razdelili na tri faze (Slika 4.18): fizikalni pogon, krmilnik zamahov in podatkovno skladišče zamahov. Fizikalni pogon krmilnika podatke letenja simulira s frekvenco 2400Hz, s čimer se zagotavlja stabilnost fizikalnega skeleta. Ker se za doseg te stabilnosti vzorci, zajeti pri 240Hz, razširijo na 2400Hz, se zaradi dopolnjevanja manjkajočih podatkov lahko pojavijo določene nezveznosti v spreminjanju kotov. Krmilnik za testiranje delovanja potrebuje dostop do poti letenja, s pomočjo katere se za doseg želenega cilja pridobi zamah iz podatkovne baze zamahov in v končni fazi posreduje fizikalnemu krmilniku.

4.4.1 Reprezentacija zamaha

Eden od večjih izzivov modelov s tako podrobnim skeletom je število prostostnih stopenj, ki je kot v tem primeru preobsežna za preiskovanje celotnega prostora stanj. Model Ju, et al. [25] to problematiko rešuje z uporabo zajetega gibanja ptice (Slika 4.19), s čemer je omogočen vpogled v način gibanja ptice in posledično lažjo izdelavo vizualno pristnejše simulacije letenja. Vodenje parametrov zamaha se izvaja podobno kot pri Wu & Popović [22], le da se tokrat namesto vektorja parov parametrov zamaha hrani \mathbf{u} kot vektor transformacijskih koeficientov skaliranja, transliranja in časovnega popačenja (Tabela

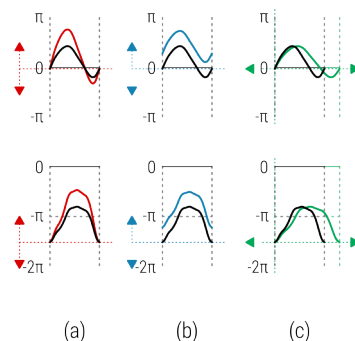
Parameter	Stran	Opis
W		Časovno popačenje
u_1^t, u_1^s	L/D	Rama kot zamahovanja
u_2^t, u_2^s	L/D	Rama kot zvijanja
u_3^t, u_3^s	L/D	Rama kot uklanjanja naprej-nazaj
u_4^t, u_4^s	L/D	Komolec kot uklanjanja naprej-nazaj
u_5^t, u_5^s	L/D	Komolec kot zvijanja
u_6^t, u_6^s	L/D	Zapestje kot zvijanja
u_7^t, u_7^s		Rep kot upogibanja
u_8^t, u_8^s		Rep kot razpiranja

Tabela 4.7 Prikaz vektorja parametrov zamaha u . Število vseh parametrov je 29, od tega eden za časovno popačitev.



Slika 4.19 Grafi spreminjanja kotov posameznih sklepov pridobljenih iz realnega gibanja goloba. (x-os čas, y-os radian).

Svetle krivulje predstavljajo gibanje leve strani ptice, temne krivulje pa predstavljajo gibanje desne strani ptice, z izjemo repa, ki je označen z zeleno barvo. Funkcije spreminjanja kotov sklepov so obarvane primerljivo funkcijama Wu & Popović [22].



Slika 4.20 Prikaz transformacije referenčnega zamaha in generiranje nove množice parametrov zamaha. (a) Skaliranje (b) Translacija (c) Časovno popačenje

4.7), potrebnih za modifikacijo referenčnega zamaha (Slika 4.20).

4.4.2 Podatkovna baza zamahov

Delovanje krmilnika je odvisno od pokritosti podatkovne baze zamahov. Polnjenje podatkovne baze, ki ga bomo razložili v nadaljevanju, poteka z uporabo nenadzorovanega učenja, na podlagi naključnih začetnih pozicij, orientacij, hitrosti in transformacijskih parametrov, ki privedejo do izdelave novega zamaha. Takšen način učenja zamahov lahko tudi privede do nastanka manevrov, ki so za pravo ptico v naravi neizvedljivi, zato je s tovrstnim učenjem potrebnega veliko ročnega prečiščevanja. Za zagotavljanje čim večje stabilnosti krmilnika mora baza vsebovati najmanj 20000 različnih zamahov. Dimenzija problema hranjenja in posledično pridobivanja pravih zamahov zahteva definiranje enoličnega najmanjšega opisa dinamičnega stanja ptice

$$\mathbb{S} = [q_{r,p}, q_{r,r}, \dot{\mathbf{q}}_p, \dot{\mathbf{q}}_r], \quad (4.41)$$

kjer $q_{r,p}$ in $q_{r,r}$ v tem vrstnem redu predstavljata kot naklona (dviganja/spušcanja) in kot nagiba (levo/desno), $\dot{\mathbf{q}}_p$ linearno hitrost in $\dot{\mathbf{q}}_r$ kotno hitrost. Zamah ostane definiran kot vektor parametrov, \mathbf{u} .

Polnjenje

Polnjenje podatkovne baze zamahov je razdeljeno na dva postopka: obogatitev naučenih podatkov in ponovno vzorčenje s prečiščevanjem. Pri obogatitvi naučenih podatkov se najprej izbere naključno začetno stanje \mathbb{S} in naključno spremembo parametrov $\Delta\mathbf{u}$, za kar se simulacija zažene dvakrat; enkrat z \mathbf{u}_0 referenčnimi parametri, v drugo pa

z $\mathbf{u}_0 + \Delta\mathbf{u}$. Razlika med končnima stanjema, $\Delta\mathbb{S}$, se shrani v bazo učnih zamahov kot $(\Delta\mathbb{S}, \Delta\mathbf{u})$. Zamahi, pridobljeni v prvi iteraciji, so le korak obogatitve učne baze zamahov in začetni korak njenega prečiščevanja. Potreba po tem postopku nastane kot posledica uporabe fizikalne simulacije (pridobivanje trenutnega stanja) in naključno generiranih stanj. To namreč privede do nastanka neenakomerno zapolnjene baze. Tako se 20000 do 100000 stanj pridobljenih v prvem koraku, ponovno uporabi kot učeno bazo zamahov, s katero se izdelava novo bazo, le z boljšo porazdelitvijo stanj. Za zagotavljanje uporabe čim bolj pristnih stanj se v končni fazi še ročno izbere relevantne regije med stanji dviganja/spuščanja levo in desno, ter odstrani najbolj odstopajoča stanja.

Zajemanje in apliciranje

Kot že omenjeno, je bistvo krmilnika, da generira naslednje stanje ptice, \mathbf{S} , z uporabo referenčnega zamaha. To pomeni, da se v dinamičnem stanju \mathbb{S}_c pridobi vektor parametrov zamaha \mathbf{u} na podlagi znanega naslednjega dinamičnega stanja, \mathbb{S}_n . Iz tega sledi, da je naloga krmilnika pridobivanje parametrov zamaha na podlagi para dinamičnih stanj $(\mathbb{S}_c, \mathbb{S}_n)$. Pri zagonu simulacije krmilnik v dinamičnem stanju \mathbb{S}_c uporabi za generiranje čisto prvega zamaha referenčne parametre, \mathbf{u}_0 . Iz dobljenega dinamičnega stanja, \mathbb{S}_0 , in z uporabo zelenega naslednjega dinamičnega stanja, \mathbb{S}_n , izračuna spremembo $\Delta\mathbb{S} = \mathbb{S}_n - \mathbb{S}_0$, s katero se iz baze zamahov pridobi korekcijo parametrov referenčnega zamaha, $\Delta\mathbf{u}$, potrebno za doseg želenega dinamičnega stanja \mathbb{S}_n .

Ker podatkovna baza ni nujno najboljše pokrita, se za pridobivanje parametrov uporabi regresijo k najbližjih sosedov. Za katero koli spremembo dinamičnega stanja $\Delta\mathbb{S}$ se razlika med dinamičnima stanjema računa z enačbo

$$d(\Delta\mathbb{S}^0, \Delta\mathbb{S}^1) = (\Delta\mathbb{S}^0 - \Delta\mathbb{S}^1)\mathbf{W}(\Delta\mathbb{S}^0 - \Delta\mathbb{S}^1)^T, \quad (4.42)$$

kjer je \mathbf{W} diagonalna matrika uteži, s katerimi se poudari pomembnost posameznega dela dinamičnega stanja ($W_{0,0} = W_{1,1} = 0.2$ za naklon in nagib, $W_{2,2} = 0.24$ za linearno hitrost ter $W_{3,3} = 0.4$ za kotno hitrost), $\Delta\mathbb{S}^0$ in $\Delta\mathbb{S}^1$ pa sta dinamični stanji med katerima se računa obtežena razlika. Sprememba parametrov se iz baze zamahov izračunana kot obtežena vsota k najbližjih sosedov

$$\Delta\mathbf{u} = \sum_{i \in kNN} \frac{d(\Delta\mathbb{S}, \Delta\mathbb{S}_i)^{-1}}{D} \Delta\mathbf{u}_i, \quad (4.43)$$

kjer je teža vsakega vzorca inverzno proporcionalna razliki želene spremembe stanja $\Delta\mathbb{S}$ in

$D = \sum_{i \in kNN} d(\Delta S, \Delta S_i)^{-1}$ vsoti vseh uteži, namenjena normalizaciji končnega vektorja korekcije parametrov referenčnega zamaha.

5 Razprava

Z izdelavo temeljitega fizikalnega krmilnika ene ptice, je delovanje celovitega modela letenja v jatah prej resničnost kot fikcija. Eden od ključnih problemov, pogojenih z omejitvami strojne opreme, se pojavi le pri zagotavljanju izvajanja simulacije v realnem času. Ta problem je prisoten zaradi zagotavljanja stabilnosti fizikalnih pogonov, kar pomeni, da je za izdelavo simulacije manjših ptic, kot so to npr. vrabci, treba izvajati simulacijo pri frekvenci vzorčenja najmanj $4000Hz$ (pri kolibrijah to lahko preide tudi do $8000Hz$).

Zaradi strojnih omejitev so fizikalni modeli tudi temu primerno izdelani, omejeni v delovanju. Kar pomeni; če je model primeren izvajanju v realnem času, je zaradi tega vizualno zelo neprepričljiv, preveč preprosto modeliran in zmožen le generiranja simetričnih zamahov (kar zmanjša število vrst ptic, možnih modeliranja), primer takega modela je model Zhu, et al. [23]. V primeru, da je model bolj dodelan, je preizkušen zgolj pri letenju naravnost z možnostjo spreminjanja vpadnega kota, [24, 36, 37]. Vseeno obstajajo modeli, ki se kljub tem preprekam, podajajo v temeljitejšo izdelavo, namenjeno dinamičnem manevriranju po prostoru. Kot najbolj smiselna za nadaljnje razširitve, sta

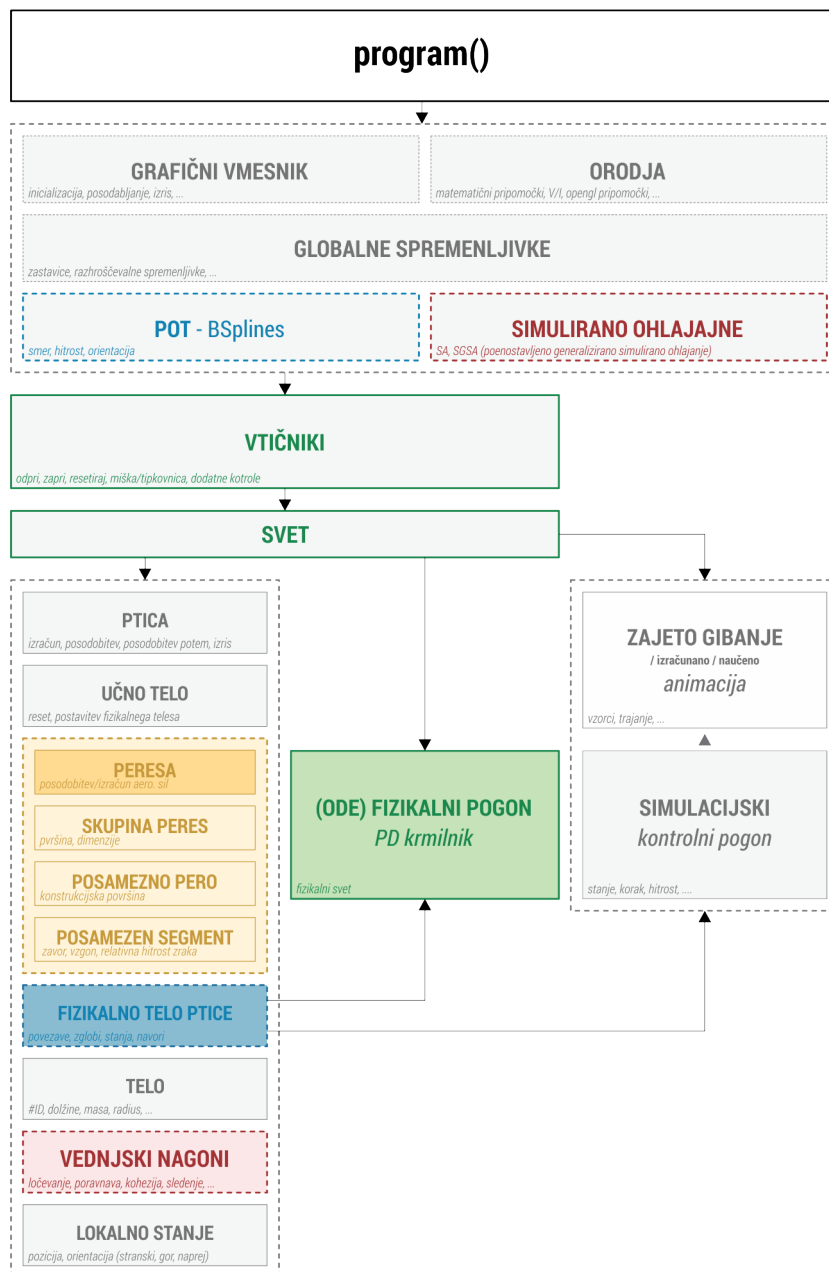
modela Wu & Popović [22] in Ju, et al. [25], ki z želijo izdelave čim bolj pristne animacije letenja, vsak na svoj način predstavljata učenje zamahov.

Izhodišče učnega modela naj bi bilo pridobivanje znanja na podlagi izkušenj. Pridobivanje znanja z generaliziranjem daje modelom možnost, da delujejo bolj natančno na novih, neznanih primerih. Iz tega sledi, da ne gre le za učenje na pamet, ampak za iskanje pravil v znanih podatkih. Glede na nadzor se “strojno” učenje deli na:

- **nadzorovano učenje** (angl. *supervised learning*): model se uči s podanimi pari vhodnih podatkov in želenih izidov; pri tem želene izhodne vrednosti določa učitelj oziroma nadzornik;
- **ne-nadzorovano učenje** (angl. *unsupervised learning*): učni model se po ocenah oziroma kriterijih glede na vhodne podatke razdeli, v kategorije po značilnostih oziroma v t. i. skupine (angl. *clusters*);
- **vzpodbujevalno učenje** (angl. *reinforcement learning*): model se uči z nagrajevanjem in kaznovanjem, ki se tudi imenuje maksimiranje uporabnosti agenta.

Pri modelih kot so Zhu, et al. [23], je nadgradnja v učni model nesmiselna, zaradi avtomatiziranega zamahovanja na podlagi v naprej izbrane točke letenja. Model, kot je Ju, et al. [25], v osnovi uporablja način nenadzorovanega učenja. To pomeni, da se na podlagi zajetih podatkov izdela množico pravih zamahov in referenčni zamah. Z referenčnim zamahom se z naključnim spreminjanjem napolni manjkajoči del še potrebnih zamahov. Dobljene podatke se razdeli v skupine in odstrani zamahe s prevelikim odstopanjem od množice pravih zamahov. Model je za izdelavo krmilnika letenja idealen, a zaradi tovrstnega načina učenja lahko kaj hitro privede do slabega pokritja podatkovne baze zamahov. S tem namenom se zato poslužuje metod učenja, katerih način je usmerjanje učnega modela k želenemu cilju. Ker so taki načini veliko bolj dolgotrajni in je z njimi težje zagotoviti pridobivanje optimalnih rešitev v najkrajšem možnem času, se za reševanje tovrstnih problemov navdih jemlje iz narave. S tem pride v poštev model, kot ga predstavlja Wu & Popović [22], ki za učenje uporablja simulirano ohlajanje pod vejo genetskih algoritmov (oziroma vzpodbujevalno učenje).

Sledeč iz tega, smo po vzoru Wu & Popović [22] razvili model ptice, predstavljen v 2. poglavju, pri katerem smo se odločili, da ukrivljanja peres ne bomo modelirali na podlagi trditve Parslewa [24], da je to zgolj vizualni dodatek. Iz nastalega modela, smo razvili



Slika 5.1 Prikaz implementiranega simulatorja, zmožnega simuliranja vedenjskih gonilnih sil, fizikalnega krmiljenja polno-funkcionalnega skeleta ptice s pomočjo zajetih zamahov in učenja le teh.

predvajalnik (Slika 5.1), zmožen krmiljenja fizikalnega skeleta ptice v realnem času, na podlagi v naprej generiranih zamahov (ali naučenih). Testiranje slednjega smo opravili na podlagi javno objavljenih zajetih podatkov gibanja ptice Ju, et al. [25]. Ker je cilj modela Wu & Popović [22] le izdelava animacije s predhodno polnjenimi zelenimi spremembami skeleta, je treba idejo združevanja zamahov razširiti do meje, zmožne simuliranja letenja v realnem času, katero podrobneje predstavimo v nadaljevanju.

Za zagotavljanje čim bolj stabilnega krmiljenja fizikalnega skeleta smo se pri našem modelu odločili uporabiti rešitev, predstavljeno s strani Tan, et al. [32], ki temelji na uporabi SPD krmilnikov, ki smo jo opisali v 3. poglavju. Za razliko od osnovne ideje računanja vzgonskega koeficienta Wu & Popović [22] uporabimo način, ki ga je predstavil Parslew [24], za katerega smo naredili korekcijo parametra A iz enačbe (4.40), tako da ta sovpada z vrednostjo L_3 iz tabele 4.3. To vrstno računanje vzgonskega koeficienta privede fizikalno animacijo do veliko boljše dinamike zamahovanja. Polna funkcionalnost krmilnika letenja v jatah pa je zagotovljena le, ko model lahko poljubno manevriramo po prostoru. S tem namenom se pojavi potreba po izdelavi primerne podatkovne baze zamahov, kot jo tudi predstavijo Ju, et al. [25]. Ker ta način ni najbolj primeren za uporabo z modelom, ki temelji na vzpodbujevalnem učenju, v nadaljevanju predstavimo tudi predlog nadgradnje podatkovne baze zamahov.

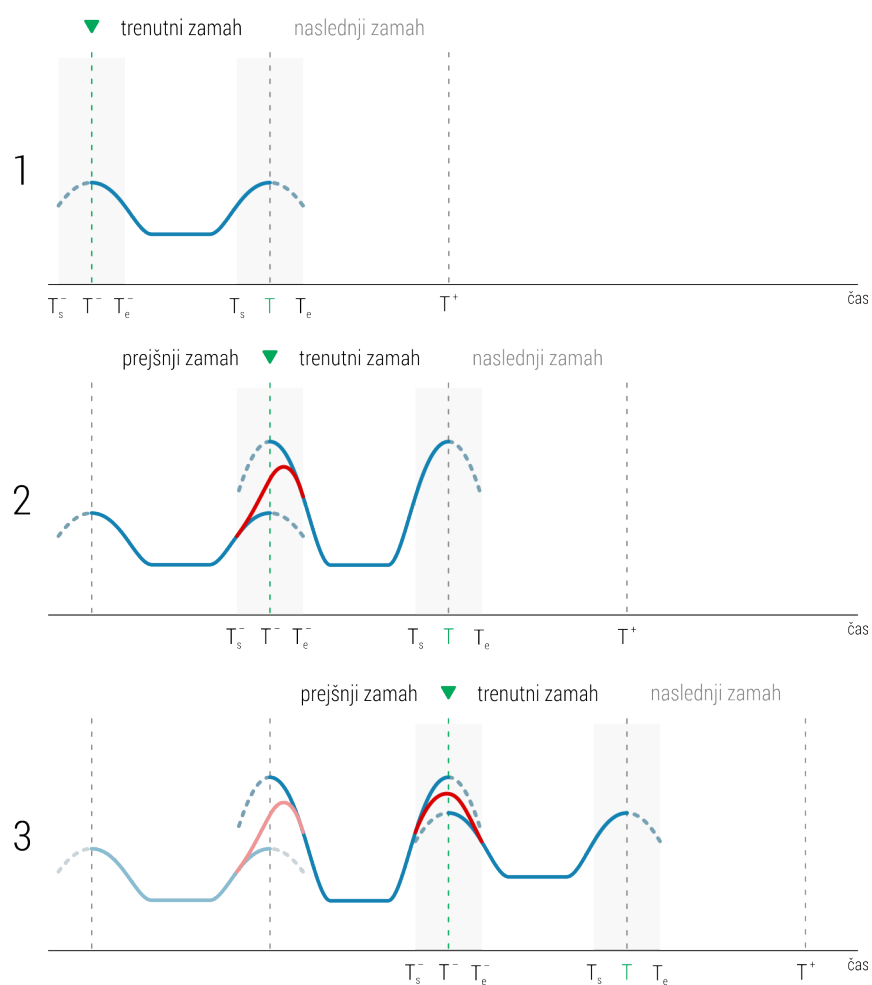
5.1 Generiranje in združevanje zamahov

Modeli, kot so Wu & Popović [22], združevanje zamahov izvajajo na podlagi hranjenja vseh sprememb v sklepih, ki se po optimizaciji uporabijo za izdelavo renderirane animacije leta. Iz tega razloga smo se odločili nadgraditi osnovno idejo do te mere, da bi omogočila izvajanje enakega postopka, le brez pred-polnjenja sprememb v vseh sklepih, tekom celotne optimizacije.

Posodobljen postopek v osnovi zahteva le hranjenje vektorjev \mathbf{u}^- , \mathbf{u} in \mathbf{u}^+ , ki so v tem vrstnem redu prejšnji parametri zamaha, trenutni parametri zamaha in naslednji parametri zamaha. Po potrebi glajenja zamahov med poljubnimi vrednostmi se osnovno gladilno funkcijo (4.11) razširi v $b_{acc}(t, t_0, t_1) = t_0 + (t_1 - t_0)b_{acc}(t)$. Za pravilno spreminjanje faze funkcij g_1 in g_2 , se poleg $\phi(t) = \frac{2\pi t}{T}$, uvede še

$$\phi^-(t) = \frac{2\pi(t + T^-)}{T^-}, \quad \phi^+(t) = \frac{2\pi(t - T)}{T^+}, \quad (5.1)$$

s katerima se v času $t \in [0, T]$, trajanju trenutnega zamaha, izračuna fazni zamik name-



Slika 5.2 Grafi prikazujejo delovanje metode za računanje zaporedja različnih zamahov brez uporabe predhodnega polnjenja vzorcev. Zelene puščice prikazujejo trenutna stanja simulacije in rdeče krivulje predstavljajo končno glajeno krivuljo med prejšnjim zamahom in trenutnim.

njen glajenju s funkcijo iz preteklosti, $\phi^-(t)$, ki je veljaven pri $t \in [0, T_e^-]$, in fazni zamik namenjen glajenju s funkcijo iz prihodnosti, $\phi^+(t)$, ki je veljaven pri $t \in [T_s, T]$. Uporabljeni parametri so: čas preteklega zamaha, T^- , trenutnega zamaha, T , in naslednjega zamaha, T^+ , ki se iz baze zamahov, v tem vrstnem redu nastavijo na podlagi \mathbf{u}^- , \mathbf{u} in \mathbf{u}^+ . Kot je vidno na sliki 5.2 je za glajenje sosednjih zamahov potrebno izračunati

$$\begin{aligned} T_s^- &= -\min(T^-, T) \frac{1}{5}, & T_e^- &= \min(T^-, T) \frac{1}{5}, \\ T_s &= T - \min(T, T^+) \frac{1}{5}, & T_e &= T + \min(T, T^+) \frac{1}{5}, \end{aligned} \quad (5.2)$$

kjer so T_s^- in T_e^- v tem vrstnem redu začetek in konec glajenja prejšnjega in trenutnega zamaha, T_s in T_e pa v tem vrstnem redu začetek in konec glajenja trenutnega in naslednjega zamaha. Glajenje med zamahi se izvaja 20% časa najkrajšega zamaha, pri čemer se gladi med obema obravnavanima zamahoma. Združevanje med sosednjimi zamahi se v predstavljenem krmilniku izvaja s postopkom:

$$q_{z,i}(t) = \underline{u}_i + (\bar{u}_i - \underline{u}_i)g_k(u_i^u, u_i^d, h^t(\phi(t), u_{\alpha|\beta}))$$

če $t < T_e^-$ potem

$$q_{z,i}^-(t) = \underline{u}_i + (\bar{u}_i - \underline{u}_i)g_k(u_i^{u-}, u_i^{d-}, h^t(\phi^-(t), u_{\alpha|\beta}^-))$$

$$q_{z,i}^*(t) = q_{z,i}^-(t) + (q_{z,i}(t) - q_{z,i}^-(t))b_{acc}(t, T_s^-, T_e^-)$$

drugače, če $t > T_s$ potem

$$q_{z,i}^+(t) = \underline{u}_i + (\bar{u}_i - \underline{u}_i)g_k(u_i^{u+}, u_i^{d+}, h^t(\phi^+(t), u_{\alpha|\beta}^+))$$

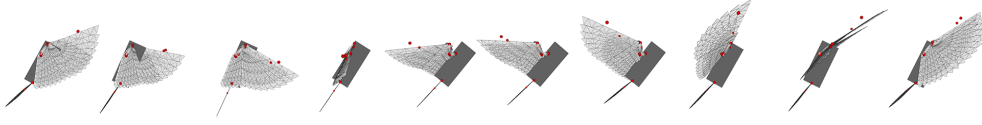
$$q_{z,i}^*(t) = q_{z,i}(t) + (q_{z,i}^+(t) - q_{z,i}(t))b_{acc}(t, T_s, T_e)$$

drugače

$$q_{z,i}^*(t) = q_{z,i}(t)$$

V omenjenem postopku se poleg $q_{z,i}^*(t)$, želenega stanja vpelje začasno vodenje trenutnega, $q_{z,i}(t)$, prejšnjega, $q_{z,i}^-(t)$, in naslednjega, $q_{z,i}^+(t)$, želenega stanja, med katerimi se izvaja rekonstrukcija celotnega seta zamahov. Pravilno delovanje slednjega, je zagotovljeno ko se ob zaključku vsakega cikla zamaha t nastavi na 0 in se ponovno izračunajo enačbe (5.2).

Dobro definirani funkciji g_1 in g_2 s strani Wu & Popović [22] omogočata tudi generiranje zamahov goloba, zajetega s strani Ju, et al. [25]. Dosega takega zamaha je mogoča le, ko se funkcijo g_2 uporabi za obe prostostni stopnji komolca in g_1 za vse ostale sklepe. S tem se izgubi avtomatizacijo zapiranja krila, a se pridobi verodostojen zamah (Slika 5.3), katerega Wu & Popović [22] nista bila zmožna zagotoviti.



Slika 5.3 Prikaz enega cikla zamaha modela ptice Wu & Popović [22] z uporabo spremenjene logike zamahovanja, pridobljene s pomočjo zajetih podatkov leta goloba s strani Ju, et al. [25].

5.2 Izdelava podatkovne baze parametrov letenja

Izdelava podatkovne baze zamahov je eden od ključnih delov krmiljenja polno funkcionalnega skeleta ptice v realnem času. Za doseganje le tega Ju, et al. [25] predstavijo način hranjenja enoličnega najmanjšega opisa dinamičnega stanja ptice, \mathbb{S} . Za možnost prejemanja pravilnega vektorja parametrov zamaha, \mathbf{u} , je dinamično stanje sestavljeno iz parametrov kota naklona, $q_{r,p}$, kota nagiba, $q_{r,r}$, linearne hitrosti, $\dot{\mathbf{q}}_p$, in kotne hitrosti, $\dot{\mathbf{q}}_r$. Slabost pristopa je pogojena s tem, da se tipi zamahov razlikujejo, ne le s spremembo orientacije ampak tudi v odvisnosti od vpadnega kota trupa, pri katerem se zamah izvaja. Iz tega sledi, da je s tovrstno bazo zamahov nemogoče zagotoviti robusten krmilnik letenja pod različnimi vpadnimi koti trupa. Da bi se izognili tej slabosti, za predstavljen model predlagamo minimalno spremembo dinamičnega stanja in logike hranjenja podatkov.

Pri izdelavi podatkovne baze, ki bi omogočala čim hitrejšo pridobivanje želenega stanja, se velikokrat predlaga implementacijo KD-Drevesa (razdeljevanje dreves v 2D prostoru s pravokotniki, v 3D s kvadri) kot enostavno rešitev ali VP-drevesa (razdeljevanje dreves v 2D prostoru s krogi, v 3D s sferami) kot optimalnejšo. Kljub temu, da se obe strukturi lahko razširi za uporabo več dimenzionalnih stanj, je pridobivanje slednjih z vsako dimenzijo dolgotrajnejše.

5.2.1 Polnjenje baze

Bazo zamahov bi se, za razliko od modela Ju, et al. [25], polnilo z uporabo enoličnega najmanjšega opisa dinamičnega stanja ptice

$$\mathbb{S} = [q_{r,p}, q_{r,r}, \dot{\mathbf{q}}_p, \dot{\mathbf{q}}_p^*], \quad (5.3)$$

kjer prvi del predstavlja stanje ob zajemu zamaha, kjer je $q_{r,p}$ naklon in $q_{r,r}$ nagib trupa ptice, relativen na koordinatni sistem sveta, drugi del trenutno stanje gibanja, kjer je $\dot{\mathbf{q}}_p$ smer leta in hitrost, zadnji del pa želeno stanje oziroma pospešek ob zaključku zamaha,

$\ddot{\mathbf{q}}_p^*$. Velik poudarek je, da se dinamično stanje ne hrani tako kot pri Ju, et al. [25], ki hranijo spremembo referenčnega zamaha, potrebno za doseg spremembe dinamičnega stanja $(\Delta\mathbf{S}, \Delta\mathbf{u})$, ampak bi se hranil kar par (\mathbf{S}, \mathbf{u}) . Edini večji korak nazaj takega hranjenja je računanje transformacij $\dot{\mathbf{q}}_p$ in $\ddot{\mathbf{q}}_p^*$ iz lokalnega koordinatnega sistema ptice v koordinatni sistem sveta in/ali obratno.

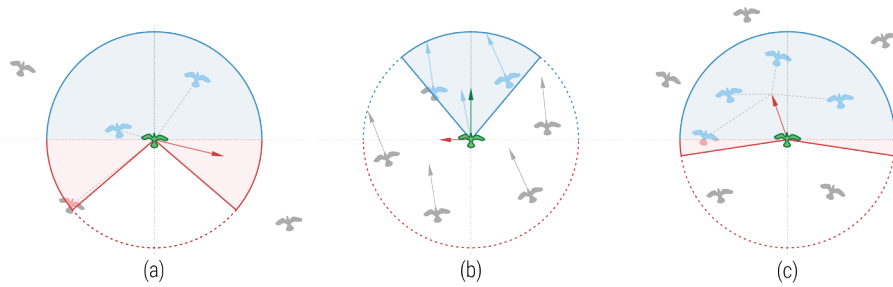
Vzpodbujevalni način učenja v prid nenadzorovanemu učenju se iz izdelane baze razdeli na dve fazi: učenje simetričnega letenja, pod različnimi vpadnimi koti, in učenje manevrskega letenja iz različnih vpadnih kotov. S tako razdelitvijo se zagotovi izgradnja zamahov za stabilnejši let naravnost in posledično dobro podlago za učenje manevrskih zamahov. V prid simetričnosti ptice levo/desno se postopek učenja manevrskih zamahov izvede le v eno smer, levo ali desno, druga stran pa je le preslikava čez ustrezno ravnino ptice (Slika 4.1).

5.2.2 Pridobivanje podatkov iz baze zamahov

Potrebe vedenjskih modelov v večjem delu temeljijo na uporabi gonilne sile, \mathbf{F}_s , (Slika 5.4), kot edini regulator, namenjen usmerjanju ptice (večinoma v odvisnosti od soseščine). Iz tega sledi, da je za izračun pospeška iz vedenjske sile treba izračunati le

$$\ddot{\mathbf{q}}_p^* = \frac{\mathbf{F}'_s}{m}, \quad (5.4)$$

kjer je \mathbf{F}'_s gonilna vedenjska sila transformirana iz lokalnega koordinatnega sistema ptice v koordinatni sistem sveta in m masa. Dobljen pospešek, $\ddot{\mathbf{q}}_p^*$, se označi kot pospešek po zaključku enega zamaha. Za smer in hitrost se uporabi trenutno hitrost ptice, $\dot{\mathbf{q}}_p$. Naklon, $q_{r,p}$, in nagib trupa, $q_{r,r}$, se izračunata na podlagi trenutne orientacije ptice \mathbf{q}_r .



Slika 5.4 Zasenčena modra/rdeča polja predstavljajo polje zaznavanja, zelena ptica je objekt za katerega se računa zaznavanje, rdeča puščica prikazuje vedenjsko silo in modre puščice zelene smeri letenja. (a) LOČEVANJE (b) PORAVNAVA (c) KOHEZIJA.

Vse dobljene vrednosti se zloži v stanje \mathbb{S} , s pomočjo katerega se iz baze zamahov pridobi vektor parametrov zamaha, \mathbf{u} , potreben za doseg želenega cilja; pospeška vedenjske sile.

5.3 Sklep

Zaradi obsežnosti problematike je, kljub implementiranemu, ostalo veliko odprtih in nerešenih problemov. Za zagotavljanje večje skalabilnosti fizikalnega pogona ob želji po čim manjšim ročnim popravkom ter odvisnosti od parametrov, bi bilo kljub uporabi SPD krmilnika potrebno pridobiti boljšo metodo računanja navorov sklepov. Z željo po zasnovi krmilnikov, ki naj ne bi bili vezani le na eno vrsto ptice, bi potrebovali tudi verodostojne in podrobnejše meritve ptic (Slika 2.8). Poudariti je potrebno, da je učni model ptice, predstavljen v nalogi, zmožen le učenja ravnega leta, kjer predvidevamo, da je vzrok odločitve, da ne bomo upoštevali ukrivljanja peres, katerih pomembnost po eni strani zavrača Parslew [24] po drugi strani pa jo izpostavljajo Ju, et al. [25]. Posledično ostaja zaradi nezmožnosti manevriranja modela ptice po prostoru, delovanje krmilnika s pomočjo podatkovne baze zamahov nepreverjeno. Implementacijo slednjega, si puščamo za nadaljnje delo.

LITERATURA

- [1] X. Cui, J. Gao, T. E. Potok, A Flocking Based Algorithm for Document Clustering Analysis, *Journal of Systems Architecture* 52 (8) (2006) 505–515.
- [2] D. J. T. Sumpter, The principles of collective animal behaviour, *Philosophical Transactions of the Royal Society B: Biological Sciences* 361 (1465) (2006) 5–22.
- [3] I. Lebar Bajec, F. H. Heppner, Organized flight in birds, *Animal Behaviour* 78 (2009) 777–789.
- [4] T. Vicsek, A. Zafeiris, Collective motion, *Physics Reports* 517 (3–4) (2012) 71–140.
- [5] A. Okubo, Dynamical aspects of animal grouping: Swarms, schools, flocks, and herds, *Advances in Biophysics* 22 (1986) 1–94.
- [6] I. Aoki, A simulation study of the schooling mechanism in fish, *Bulletin of the Japanese Society of Scientific Fisheries* 48 (8) (1982) 1081–1088.
- [7] C. W. Reynolds, Computer Animation in the World of Actors and Scripts, MSc thesis, Massachusetts Institute of Technology, The Architecture Machine Group, Cambridge, MA (1978).
- [8] C. W. Reynolds, Flocks, herds and schools: A distributed behavioral model, *ACM SIGGRAPH Computer Graphics* 21 (4) (1987) 25–34.
- [9] D. Pangeršič, Računalniški model usklajenega pristajanja in vzletanja, BSc thesis, Fakulteta za računalništvo in informatiko, Ljubljana, Slovenia (2012).
- [10] C. K. Hemelrijk, H. Hildenbrandt, Schools of fish and flocks of birds: their shape and internal structure by self-organization, *Interface Focus* 2 (6) (2012) 726–737.
- [11] X. Wang, X. Jin, Z. Deng, L. Zhou, Inherent noise-aware insect swarm simulation, *Computer Graphics Forum* 33 (6) (2014) 51–62.

- [12] J.-M. Lien, O. B. Bayazit, R. T. Sowell, S. Rodriguez, N. M. Amato, *Shepherding behaviors*, in: *Proceedings of the International Conference on Robotics and Automation (ICRA)*, Vol. 4, 2004, pp. 4159–4164.
- [13] M. Barksten, D. Rydberg, *Extending reynolds' flocking model to a simulation of sheep in the presence of a predator*, Tech. rep., KTH, School of Computer Science and Communication (CSC), SE-100 44 Stockholm SWEDEN (12 2013).
- [14] D. Strömbom, R. P. Mann, A. M. Wilson, S. Hailes, A. J. Morton, D. J. T. Sumpter, A. J. King, *Solving the shepherding problem: heuristics for herding autonomous, interacting agents*, *Journal of The Royal Society Interface* 11 (100) (2014) 20140719.
- [15] I. D. Couzin, J. Krause, R. James, G. D. Ruxton, N. R. Franks, *Collective Memory and Spatial Sorting in Animal Groups*, *Journal of Theoretical Biology* 218 (2002) 1–11.
- [16] M. Ballerini, N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, A. Orlandi, G. Parisi, A. Procaccini, M. Viale, V. Zdravkovic, *Empirical investigation of starling flocks: a benchmark study in collective animal behaviour*, *Animal Behaviour* 76 (1) (2008) 201–215.
- [17] M. Ballerini, N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, V. Lecomte, A. Orlandi, G. Parisi, A. Procaccini, M. Viale, V. Zdravkovic, *Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study*, *Proceedings of The National Academy of Sciences* 105 (2008) 1232–1237.
- [18] I. Lebar Bajec, N. Zimic, M. Mraz, *Simulating flocks on the wing: the fuzzy approach*, *Journal of Theoretical Biology* 233 (2) (2005) 199–220.
- [19] I. Lebar Bajec, *Fuzzy Model for a Computer Simulation of Bird Flocking*, PhD dissertation, University of Ljubljana, Faculty of Computer and Information Science, Ljubljana, Slovenia (2005).
- [20] H. Hildenbrandt, C. Carere, C. Hemelrijk, *Self-organized aerial displays of thousands of starlings: a model*, *Behavioral Ecology* 21 (6) (2010) 1349–1359.

- [21] J. Shopf, J. Barczak, C. Oat, N. Tatarchuk, March of the Froblins: simulation and rendering massive crowds of intelligent and detailed creatures on GPU, *SIGGRAPH 2008* (2008) 52–101.
- [22] J. Wu, Z. Popović, Realistic modeling of bird flight animations, *ACM Transactions on Graphics* 22 (3) (2003) 888–895.
- [23] C. Zhu, K. Muraoka, T. Kawabata, C. Cao, T. Fujimoto, N. Chiba, Real-time Animation of Bird Flight Based on Aerodynamics, *The Journal of The Society for Art and Science* 5 (1) (2006) 1–10.
- [24] B. Parslew, Simulating avian wingbeats and wakes, PhD dissertation, University of Manchester, Faculty of Engineering and Physical Sciences, Manchester, UK (2012).
- [25] E. Ju, J. Won, J. Lee, B. Choi, J. Noh, M. G. Choi, Data-driven Control of Flapping Flight, *ACM Trans. Graph.* 32 (5) (2013) 151–163.
- [26] C. Pennycuik, Modelling the Flying Bird, Theoretical Ecology Series, Elsevier Science, 2008.
- [27] U. Norberg, Vertebrate flight: mechanics, physiology, morphology, ecology and evolution, Springer-Verlag, 1990.
- [28] F. A. Jenkins, K. P. Dial, G. E. Goslow, A Cineradiographic Analysis of Bird Flight: The Wishbone in Starlings Is a Spring, *Science* 241 (4872) (1988) 1495–1498.
- [29] B. Tobalske, K. Dial, Flight kinematics of black-billed magpies and pigeons over a wide range of speeds, *Journal of Experimental Biology* 199 (2) (1996) 263–280.
- [30] R. H. J. Brown, The flight of birds: The flapping cycle of the pigeon, *Journal of Experimental Biology* 25 (4) (1948) 322–333.
- [31] T. Geijtenbeek, N. Pronost, Interactive Character Animation Using Simulated Physics: A State-of-the-Art Review, *Comput. Graph. Forum* 31 (8) (2012) 2492–2515.
- [32] J. Tan, K. Liu, G. Turk, Stable proportional-derivative controllers, *IEEE Computer Graphics and Applications* 31 (4) (2011) 34–44.

- [33] B. Ramakrishnananda, K. C. Wong, Animating bird flight using aerodynamics, *The Visual Computer* **15** (1999) 494–508.
- [34] J.-H. Lee, M.-S. Kim, Pseudo dynamic keyframe animation with motion blending on the configuration space of a moving mechanism, in: *Computer Graphics and Applications*, S.Y. Shin and T.L. Kunii (Eds.), World Scientific, Vol. 3, World Scientific, 1995, pp. 118–132.
- [35] H. Wang, J. Kearney, K. Atkinson, Robust and efficient computation of the closest point on a spline curve, in: *In Proceedings of the 5th International Conference on Curves and Surfaces*, 2002, pp. 397–406.
- [36] B. Parslew, Low Order Modelling of Flapping Wing Aerodynamics for Real-Time Model Based Animation of Flapping Flight, MSc thesis, University of Manchester, Faculty of Engineering and Physical Sciences, Manchester, UK (2005).
- [37] B. Parslew, W. J. Crowther, Simulating avian wingbeat kinematics, *Journal of Biomechanics* **43** (16) (2010) 3191–3198.
- [38] E. Ju, B. Choi, J. Noh, J. Lee, Data-driven Bird Simulation, in: *ACM SIGGRAPH 2011 Talks, SIGGRAPH '11*, ACM, New York, NY, USA, 2011, p. 63.